

**IMPLEMENTASI SISTEM PEMANTAUAN KONDISI
RUANGAN MENGGUNAKAN JARINGAN SENSOR NIRKABEL
BERBASIS PROTOKOL RF24NETWORK**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
AHMAT CHOERI
NIM: 115090607111021



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI SISTEM PEMANTAUAN KONDISI RUANGAN MENGGUNAKAN
JARINGAN SENSOR NIRKABEL BERBASIS PROTOKOL RF24NETWORK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Ahmat Choeri
NIM: 115090607111021

Skripsi ini telah diuji dan dinyatakan lulus pada
03 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Adhitya Bhawiyuga, S.Kom, M.Sc
NIP: 19890720 201803 1 002

Dosen Pembimbing II



Mahendra Data, S.Kom, M.Kom
NIK: 201503 861117 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).



Malang, 03 Agustus 2018



Ahmat Choeri

NIM: 115090607111021

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan Rahmat, Karunia dan Hidayah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi ini. Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW.

Penulisan skripsi dengan judul **“Implementasi Sistem Pemantauan Kondisi Ruang Menggunakan Jaringan Sensor Nirkabel Berbasis Protokol RF24Network”** disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya. Penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan, serta dukungan dari berbagai pihak. pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih sebesar-besarnya kepada :

1. Bapak Adhitya Bhawiyuga, S.Kom, M.Sc dan Bapak Mahendra Data, S.Kom, M.Kom selaku dosen pembimbing I dan dosen pembimbing 2 yang selalu sabar memberikan bimbingan dan nasehat untuk penulis dalam menyelesaikan skripsi ini,
2. Kepada kedua orang tua, adik serta kerabat-kerabat penulis yang selalu memberikan motivasi, jasa, serta doa yang tidak pernah berhenti kepada penulis,
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya,
4. Bapak Prasetyo Iskandar, S.T dan Ibu Wiwin Lukitohadi, S.H, S.Psi, CHRM selaku staff Bimbingan Konseling yang selalu menyemangati dan memotivasi penulis dalam pengerjaan skripsi.
5. Seluruh Civitas Akademika Teknik Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi dan selama penyelesaian tugas akhir ini,
6. Teman-teman Informatika angkatan 2011 yang selalu mendukung dan berbagi ilmu dari awal perkuliahan sampai tahap akhir penyelesaian skripsi.
7. Dan kepada semua pihak yang tidak dapat penulis sebutkan satu per satu yang terlibat baik secara langsung maupun tidak langsung demi terselesaikannya tugas akhir ini.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini masih banyak kekurangan dan masih jauh dari sempurna. Untuk itu, saran dan kritik yang membangun sangat penulis harapkan. Semoga tugas akhir ini membawa manfaat bagi penyusun maupun pihak lain yang menggunakannya.

Malang, 03 Agustus 2018

ahmatchoeri1@gmail.com

ABSTRAK

Ahmat Choeri, Implementasi Sistem Pemantauan Kondisi Ruangan Menggunakan Jaringan Sensor Nirkabel Berbasis Protokol RF24Network.

Dosen Pembimbing : Adhitya Bhawiyuga, S.T, M.Sc dan Mahendra Data, S.Kom, M.Kom.

Jaringan Sensor Nirkabel (JSN) adalah infrastruktur yang terdiri dari sejumlah sensor *node* yang didistribusikan pada lingkungan tertentu dan dikoordinasikan menggunakan jaringan nirkabel. JSN dapat diimplementasikan menggunakan modul nRF24L01. Salah satu protokol yang mendukung modul nRF24L01 adalah protokol RF24. Namun untuk implementasi pemantauan kondisi ruangan di area yang luas protokol ini mempunyai kelemahan yang menggunakan mekanisme *single-hop*. Sehingga untuk jangkauan area yang luas dibutuhkan mekanisme *multi-hop*. Protokol RF24Network merupakan protokol yang dirancang untuk memenuhi kebutuhan mekanisme *multi-hop* berbasis RF24, pengguna cukup membangun header yang berisi alamat tujuan yang tepat dan jaringan akan meneruskannya ke *node* tujuan. Sesuai dengan masalah di atas, penulis melakukan implementasi sensor *node* menggunakan Arduino Nano sebagai Mikrokontroler, nRF24L01 untuk media komunikasi dan DHT11 untuk mengakuisisi data sensor. Dari hasil implementasi didapatkan kinerja jaringan untuk pemantauan kondisi ruangan dengan tingkat akurasi pengiriman data dan *delay* sebagai berikut. Mekanisme *single-hop* pengiriman data tanpa penghalang jarak 3-12 meter akurasi 100%, 15 meter 70%, 18 meter 40%, 21 meter 0% dengan total rata-rata *delay* 145 ms, Dengan penghalang jarak 3-9 meter akurasi 100%, 12 meter 60%, 15 meter 30%, 18 meter 0% dengan total rata-rata *delay* 276 ms. Mekanisme *multi-hop* tingkat akurasi pengiriman data mencapai 100% dengan rata-rata *delay* (1 *hop* tanpa penghalang 110 ms dengan jarak total 24 meter, dengan penghalang 270 ms dengan jarak total 18 meter. 2 *hop* tanpa penghalang 270 ms dengan jarak total 36 meter, dengan penghalang 350 ms dengan jarak total 27 meter. 3 *hop* tanpa penghalang 280 ms dengan jarak total 48 meter, dengan penghalang 260 ms dengan jarak total 36 meter).

Kata Kunci : JSN, RF24Network, nRF24L01, *Packet Loss*, *Delay*. Pemantauan Ruangan.

ABSTRACT

Ahmat Choeri, Implementation of Room Condition Monitoring System Using Wireless Sensor Network Based on RF24Network Protocol.

Advisor: Adhitya Bhawiyuga, S.Kom, M.Sc and Mahendra Data, S.Kom, M.Kom

Wireless Sensor Network (WSN) is an infrastructure consisting of a number of sensor nodes that are distributed in certain environments and coordinated using a wireless network. WSN can be implemented using the nRF24L01 module. One protocol that supports the nRF24L01 module is the RF24 protocol. But for the implementation of monitoring the vast room conditions this protocol has the disadvantage of using a single-hop mechanism. So that for a large area range, multi-hop mechanism is needed. The RF24Network protocol is a protocol designed to meet the needs of RF24-based multi-hop mechanisms, users simply build a header that contains the correct destination address and the network will forward it to the destination node. In accordance with the above problems, the authors implement sensor nodes using Arduino Nano as Microcontroller, nRF24L01 for communication media and DHT11 to acquire sensor data. From the implementation results obtained network performance for monitoring the condition of the room with the level of accuracy of data transmission and delay as follows. Single-hop data transmission mechanism without barrier distance 3-12 meters 100% accuracy, 15 meters 70%, 18 meters 40%, 21 meters 0% with a total average delay of 145 ms, with a distance barrier of 3-9 meters 100% accuracy, 12 meters 60%, 15 meters 30%, 18 meters 0% with a total average delay of 276 ms. The multi-hop mechanism of data transmission accuracy reaches 100% with average delay (1 hop without barrier 110 ms with a total distance of 24 meters, with a barrier of 270 ms with a total distance of 18 meters. 2 hops without barrier 270 ms with a total distance of 36 meters, with a 350 ms barrier with a total distance of 27 meters, 3 hops without a 280 ms barrier with a total distance of 48 meters, with a barrier of 260 ms with a total distance of 36 meters).

Keywords : WSN, RF24Network, nRF24L01, Paket Loss, Delay, Room Monitoring.

DAFTAR ISI

PENGESAHAN	i
PERNYATAAN ORISINALITAS	Error! Bookmark not defined.
KATA PENGANTAR.....	iii
ABSTRAK.....	iv
ABSTRACT	v
DAFTAR ISI	vi
DAFTAR TABEL.....	viii
DAFTAR GAMBAR.....	ix
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	2
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Dasar Teori	4
2.2.1 Jaringan Sensor Nirkabel.....	4
2.2.2 Arduino Nano	5
2.2.3 Arduino IDE	7
2.2.4 Modul nRF24L01	7
2.2.5 RF24Network.....	9
2.2.6 DHT11.....	11
2.2.7 Pemrograman Python	11
2.2.8 Pengukuran Jaringan Nirkabel	11
BAB 3 METODOLOGI	13
3.1 Studi Literatur.	13
3.2 Analisis Kebutuhan Sistem.....	14
3.3 Perancangan Sistem.....	14
3.3.1 Perancangan Perangkat Keras	15

3.3.2 Perancangan Perangkat Lunak.....	15
3.4 Implementasi	16
3.4.1 Implementasi Perangkat Keras	16
3.4.2 Implementasi Perangkat Lunak.....	16
3.5 Pengujian dan Analisis	16
3.5.1 Pengujian.....	17
3.5.2 Analisis	17
3.6 Kesimpulan dan Saran	17
BAB 4 REKAYASA KEBUTUHAN.....	18
4.1 Kebutuhan Fungsional	18
4.2 Kebutuhan Non Fungsional.....	18
4.2.1 Kebutuhan Perangkat Keras.....	18
4.2.2 Kebutuhan Perangkat Lunak	19
BAB 5 PERANCANGAN DAN IMPLEMENTASI	20
5.1 Perancangan Sistem.....	20
5.1.1 Topologi Jaringan	20
5.1.2 Perancangan Perangkat Keras	20
5.1.3 Perancangan Perangkat Lunak.....	25
5.2 Implementasi Sistem	28
5.2.1 Implementasi Perangkat Keras	29
5.2.2 Implementasi Perangkat Lunak.....	30
BAB 6 PENGUJIAN DAN ANALISIS.....	36
6.1 Pengujian Perangkat Keras	36
6.2 Skenario Pengujian	37
6.2.1 Pengujian Komunikasi 2 Node (<i>Single-hop</i>)	37
6.2.2 Pengujian Mekanisme <i>Multi-hop</i>	40
BAB 7 PENUTUP	55
7.1 Kesimpulan.....	55
7.2 Saran	56
DAFTAR PUSTAKA.....	57

DAFTAR TABEL

Tabel 2.1 Spesifikasi Arduino Nano.....	6
Tabel 2.2 Fungsi Pin Pada Modul nRF24L01	8
Tabel 2.3 Kategori <i>Delay</i>	12
Tabel 2.4 Kategori Degradasi <i>Packet loss</i>	12
Tabel 5.1 Koneksi pin Arduino Nano dengan nRF24L01	23
Tabel 5.2 Koneksi Pin Arduino Nano dengan DHT11	23
Tabel 5.3 Koneksi Pin Arduino Nano dengan nRF24L01	24
Tabel 6.1 <i>Packet loss</i> Mekanisme <i>Single-hop</i> Tanpa Penghalang.....	38
Tabel 6.2 <i>Packet loss</i> Mekanisme <i>Single-hop</i> dengan Penghalang.....	39
Tabel 6.3 <i>Paket Loss 1 Hop</i> Tanpa Penghalang.....	41
Tabel 6.4 <i>Packet loss 1 Hop</i> dengan Penghalang.....	44
Tabel 6.5 <i>Packet loss 2 Hop</i> Tanpa Penghalang	46
Tabel 6.6 <i>Packet loss 2 Hop</i> dengan Penghalang	48
Tabel 6.7 <i>Packet loss 3 Hop</i> Tanpa Penghalang	50
Tabel 6.8 <i>Packet loss 3 Hop</i> dengan Penghalang	53

DAFTAR GAMBAR

Gambar 2.1 Komponen Sensor <i>Node</i>	5
Gambar 2.2 Board Arduino Nano.....	6
Gambar 2.3 Interface Arduino IDE.....	7
Gambar 2.4 Modul nRF24L01	8
Gambar 2.5 Topologi RF24Network.....	9
Gambar 2.6 Pengalamatan Pada RF24Network.....	10
Gambar 2.7 Sensor DHT11	11
Gambar 3.1 Diagram Alir Pengerjaan Penelitian	13
Gambar 3.2 Diagram blok perancangan sistem.....	15
Gambar 3.3 Diagram Alir Pengujian dan Analisis.....	17
Gambar 5.1 Topologi Jaringan	20
Gambar 5.2 Diagram Blok Sistem	21
Gambar 5.3 Diagram Blok Komunikasi <i>Single-hop</i>	21
Gambar 5.4 Diagram Blok <i>Multi-hop</i>	21
Gambar 5.5 <i>Node Transmitter</i>	22
Gambar 5.6 <i>Node Receiver</i>	24
Gambar 5.7 Perancangan Perangkat Lunak <i>Node Transmitter</i>	26
Gambar 5.8 Perancangan Perangkat Lunak <i>Node Receiver</i>	27
Gambar 5.9 Diagram Alir Komunikasi Serial.	28
Gambar 5.10 Implementasi <i>Node Transmitter</i>	29
Gambar 5.11 Implementasi <i>Node Receiver</i>	30
Gambar 6.1 Pengujian Perangkat Keras.....	36
Gambar 6.2 Output dari Pengujian Perangkat Keras.....	36
Gambar 6.3 Denah Pengujian 2 <i>Node</i> Tanpa Penghalang	37
Gambar 6.4 Rata-rata <i>Delay</i> Mekanisme <i>Single-hop</i> Tanpa Penghalang	38
Gambar 6.5 Denah Pengujian 2 <i>Node</i> dengan Penghalang	39
Gambar 6.6 Rata-rata <i>Delay</i> Mekanisme <i>Single-hop</i> dengan Penghalang	40
Gambar 6.7 Denah Pengujian Mekanisme 1 <i>Hop</i> Tanpa Penghalang	41
Gambar 6.8 Mekanisme 1 <i>Hop</i> Tanpa Penghalang.....	41
Gambar 6.9 <i>Delay</i> Paket 1 <i>Hop</i> Tanpa Penghalang.....	42
Gambar 6.10 Denah Pengujian 1 <i>Hop</i> dengan Penghalang	43

Gambar 6.11 Mekanisme 1 <i>Hop</i> dengan Penghalang.....	44
Gambar 6.12 <i>Delay</i> Paket 1 <i>Hop</i> dengan Penghalang	44
Gambar 6.13 Denah Pengujian 2 <i>Hop</i> Tanpa Penghalang	45
Gambar 6.14 <i>Delay</i> Paket 2 <i>Hop</i> Tanpa Penghalang	46
Gambar 6.15 Denah Pengujian 2 <i>Hop</i> dengan Penghalang	47
Gambar 6.16 Mekanisme 2 <i>Hop</i> dengan Penghalang.....	48
Gambar 6.17 <i>Delay</i> Paket 2 <i>Hop</i> dengan Penghalang.	48
Gambar 6.18 Denah Pengujian 3 <i>Hop</i> Tanpa Penghalang	49
Gambar 6.19 Mekanisme 3 <i>Hop</i> Tanpa Penghalang.....	50
Gambar 6.20 <i>Delay</i> Paket 3 <i>Hop</i> Tanpa Penghalang	51
Gambar 6.21 Denah Pengujian 3 <i>Hop</i> dengan Penghalang	52
Gambar 6.22 Mekanisme 3 <i>Hop</i> dengan Penghalang.....	53
Gambar 6.23 <i>Delay</i> Paket 3 <i>Hop</i> dengan Penghalang	54



BAB 1 PENDAHULUAN

1.1 Latar belakang

Jaringan Sensor Nirkabel (JSN) adalah infrastruktur jaringan yang terdiri dari sejumlah sensor *node* yang didistribusikan dalam suatu lingkungan tertentu dan dikoordinasikan oleh sebuah sistem menggunakan jaringan nirkabel. Setiap *node* memiliki kemampuan untuk pengolahan data (Mikrokontroler, CPU), memiliki memori (program, data), RF *transceiver*, catu daya (baterai), dan melibatkan satu atau lebih sensor serta aktuator (Syafriil, 2013).

Teknologi Jaringan Sensor Nirkabel (JSN) banyak memberikan inspirasi dalam penerapan dan penggunaannya di segala bidang. Salah satunya adalah monitoring lingkungan. Terbukti dengan banyaknya penelitian terkait monitoring yang menggunakan teknologi JSN diantaranya yang dilakukan oleh Suhada mengenai implementasi sistem keamanan gedung menggunakan NRF24 (Suhada, 2016). Penelitian tersebut membahas mengenai pemantauan seluruh ruangan untuk mendeteksi gerakan, jika di ruangan terdapat gerakan sistem akan mengirimkan data ke *smart phone*.

Jaringan Sensor Nirkabel (JSN) banyak diimplementasikan menggunakan *ZigBee*. Selain menggunakan *Zigbee*, Jaringan Sensor Nirkabel (JSN) juga dapat dibangun menggunakan modul nRF24L01. Salah satu protokol yang mendukung modul nRF24L01 adalah protokol RF24. Namun untuk implementasi pemantauan kondisi ruangan di area yang luas protokol ini mempunyai kelemahan yang menggunakan mekanisme *single-hop*. Dikutip dari kesimpulan penelitian yang dilakukan oleh suhada, pada mekanisme *single-hop receiver* mampu menerima data maksimal 10 meter (Suhada, 2016). Sehingga untuk jangkauan area yang luas dibutuhkan mekanisme *multi-hop*.

Protokol RF24Network merupakan protokol yang di rancang untuk memenuhi kebutuhan mekanisme *multihop* berbasis RF24, pengguna cukup membangun header yang berisi alamat tujuan yang tepat dan jaringan akan meneruskannya ke *node* tujuan. Radio dalam jaringan ini dihubungkan oleh alamat yang di tugaskan ke *pipe*. Setiap radio dapat mendengarkan 6 alamat pada 6 *pipe*, oleh karena itu setiap radio memiliki *pipe* induk dan 5 *pipe* anak, yang digunakan untuk membentuk *struktur tree*.

Berdasarkan latar belakang tersebut penulis melakukan penelitian untuk "Implementasi Sistem Pemantauan Kondisi Ruangan Menggunakan Jaringan Sensor Nirkabel Berbasis Protokol RF24Network". Arduino Nano digunakan sebagai Mikrokontroler, nRF24L01 sebagai media *transmitter* dan *receiver* data, dan DHT11 untuk sensor suhu dan kelembaban. Diharapkan dari implementasi yang dilakukan, *node* sensor dapat mengakuisisi data suhu dan kelembaban ruangan dan mengirimkannya ke *node receiver* menggunakan mekanisme *multi-hop*. Penelitian ini diharapkan dapat memberikan referensi untuk pengembangan Jaringan Sensor Nirkabel (JSN) kedepannya.

1.2 Rumusan masalah

Berdasarkan uraian latar belakang di atas didapatkan susunan rumusan masalah pada penelitian ini sebagai berikut:

1. Bagaimana merancang modul JSN menggunakan Arduino Nano, nRF24L01 dan DHT11 dengan protokol RF24Network?
2. Bagaimana mekanisme pengiriman data sensor menggunakan jaringan sensor nirkabel berbasis RF24Network?
3. Bagaimana kinerja dari protokol RF24Network untuk pemantauan kondisi ruangan?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Merancang modul JSN menggunakan Arduino Nano, nRF24L01 dan DHT11 dengan protokol RF24Network.
2. Menjelaskan mekanisme pengiriman data sensor menggunakan jaringan sensor nirkabel berbasis RF24Network.
3. Mengetahui kinerja dari protokol RF24Network untuk pemantauan kondisi ruangan.

1.4 Manfaat

Penelitian ini diharapkan mempunyai manfaat untuk berbagai pihak baik bagi pembaca maupun penulis. Adapun manfaat dari penelitian ini sebagai berikut:

1. Mendapatkan wawasan mengenai jaringan sensor nirkabel.
2. Mengetahui wawasan mengenai protokol RF24Network.
3. Mendapatkan pengetahuan mengenai mekanisme pengiriman data sensor menggunakan jaringan sensor nirkabel berbasis protokol RF24Network.
4. Mengetahui kinerja dari protokol RF24Network untuk pemantauan kondisi ruangan berdasarkan parameter uji *packet loss* dan *delay*.

1.5 Batasan masalah

Ruang lingkup pembahasan dalam penelitian ini meliputi hal-hal sebagai berikut:

1. Menggunakan Arduino Nano, nRF24L01 dan DHT11
2. Pengukuran performa yang dilakukan hanya berdasarkan parameter yang telah ditentukan yaitu *packet loss* dan *delay*.
3. Penelitian dilakukan hanya menggunakan 5 modul wireless.

1.6 Sistematika pembahasan

Sistematika penulisan penelitian dari penyusunan tugas akhir secara garis besar meliputi beberapa bab, yaitu sebagai berikut:

BAB I : PENDAHULUAN

Menjelaskan mengenai latar belakang penelitian, rumusan masalah yang harus dipecahkan, batasan masalah penelitian, tujuan, manfaat serta sistematika penulisan.

BAB II : LANDASAN KEPUSTAKAAN

Menjelaskan mengenai dasar teori, literatur ilmiah yang berkaitan dengan penelitian.

BAB III : METODOLOGI

Menjelaskan mengenai metode, studi literatur, perancangan sistem, implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV : REKAYASA KEBUTUHAN

Menjelaskan mengenai kebutuhan dari sistem, baik kebutuhan yang bersifat pokok maupun kebutuhan yang bersifat opsional agar nantinya sistem dalam berjalan dengan baik.

BAB V : PERANCANGAN DAN IMPLEMENTASI SISTEM

Membahas tentang perancangan serta implementasi sistem agar nantinya sistem dapat berjalan dan bekerja dengan baik sesuai dengan arah penelitian.

BAB VI : PENGUJIAN DAN ANALISIS

Menjelaskan mengenai ruang lingkup pengujian serta memberikan hasil pengujian yang diperoleh dari pengiriman paket data berdasarkan skenario yang telah dirancang. Hasil yang ditunjukkan disertai dengan analisis kinerja berdasarkan parameter uji yang sudah ditentukan.

BAB VII : PENUTUP

Berisi kesimpulan yang didapat dari pembuatan dan pengujian serta saran yang bertujuan untuk pengembangan pada penelitian lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini membahas tentang kajian pustakan dan dasar teori yang diperlukan. Kajian pustaka membahas tentang penelitian-penelitian yang telah ada. Dasar teori membahas mengenai teori-teori yang diperlukan untuk menyusun penelitian.

2.1 Kajian Pustaka

Penelitian mengenai Jaringan Sensor Nirkabel untuk monitoring telah banyak dilakukan, di antaranya adalah penelitian yang dilakukan oleh Saputra dengan judul penelitian “Implementasi Arduino dan nRF24L01 Sebagai Modul Jaringan Sensor Nirkabel Menggunakan Metode *Master-Slave*” (Saputra, 2015). Penelitian tersebut membahas mengenai implementasi Jaringan Sensor Nirkabel menggunakan Arduino Pro Mini dan nRF24L01 dengan evaluasi kinerja berdasarkan *packet loss* dan RTT, pada penelitian tersebut modul nRF24L01 digunakan untuk komunikasi pengiriman data antar nodenya, dan protokol yang digunakan adalah mirf.

Pada penelitian yang dilakukan oleh Suhada dengan judul penelitian “Sistem Keamanan Gedung Berbasis Jaringan Sensor Nirkabel Dengan Modul NRF24” (Suhada, 2016). Penelitian tersebut membahas mengenai implementasi Jaringan Sensor Nirkabel untuk sistem keamanan gedung menggunakan modul *transciever* NRF24. Jika di ruangan terdapat gerakan *node* sensor akan mengirimkan data ke *smartphone*, dan jarak terbaik untuk pengiriman data menggunakan NRF24 adalah 1000cm (10 meter) tanpa halangan untuk mekanisme *single-hop*.

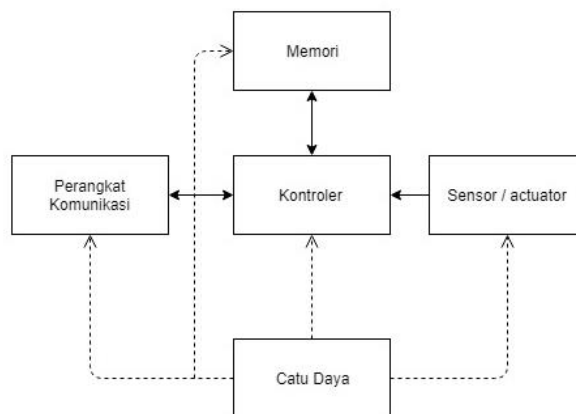
Berdasarkan referensi di atas didapat topik pembahasan menyangkut penelitian yang akan dilakukan, yaitu “Implementasi Sistem Pemantauan Kondisi Ruangan Menggunakan Jaringan Sensor Nirkabel Berbasis Protokol RF24Network”

2.2 Dasar Teori

Berdasarkan beberapa informasi yang didapat dari kajian pustaka, dalam “Implementasi Sistem Pemantauan Kondisi Ruangan Menggunakan Jaringan Sensor Nirkabel Berbasis Protokol RF24Network” mempunyai dasar teori sebagai berikut :

2.2.1 Jaringan Sensor Nirkabel

Jaringan Sensor Nirkabel (JSN) adalah infrastruktur jaringan yang terdiri dari sejumlah *node* sensor yang didistribusikan dalam suatu lingkungan tertentu dan dikoordinasikan oleh sebuah sistem menggunakan jaringan nirkabel (Setyawan, 2010). Peletakan *node* sensor dapat ditentukan maupun secara acak, sesuai dengan kebutuhan. Sebuah *node* sensor terdiri dari kontroler, sensor/actuator, memori, perangkat komunikasi dan power supply (Sugiarto, 2010). Gambar 2.1 menggambarkan sebuah sensor *node* beserta komponen pendukungnya.



Gambar 2.1 Komponen Sensor *Node*

Sumber: Sugiarto (2010)

1. Mikrokontroler (kontroler), berfungsi untuk melakukan fungsi perhitungan, mengontrol dan memproses device-device yang terhubung dengan mikrokontroler.
2. Sensor, berfungsi untuk men-sensing besaran-besaran fisis yang hendak diukur dari lingkungan.
3. Perangkat komunikasi (*transceiver*), berfungsi untuk mengirim atau menerima data kepada device lain menggunakan jaringan nirkabel.
4. External Memory, berfungsi sebagai tambahan memory bagi sistem.
5. Power Souce, berfungsi sebagai sumber energi untuk *node* sensor.

Node-node pada Jaringan Sensor Nirkabel umumnya dibedakan menjadi dua yaitu *source node* dan *sink node* (Hendrajaya, 2010). *Source node* merupakan *node* sensor yang berfungsi sebagai sumber data, yang menangkap, mengolah dan mentransmisikannya ke *node* lain. *Node sink*, yaitu *node* yang menerima data, dapat berupa sensor atau data lainnya.

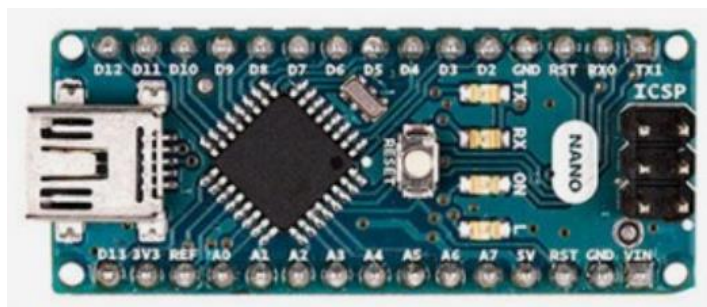
Berdasarkan interaksi antara *node source* dan *sink*, Jaringan Sensor Nirkabel dapat dikategorikan menjadi jaringan *single-hop* dan *multi-hop* (Hendrajaya, 2010).

- a. Jaringan *single-hop* terdiri dari *node* yang berkomunikasi dengan *node* lain secara langsung, tanpa melalui *node* tertentu atau *node relay*.
- b. Jaringan *multi-hop* terdiri dari *node* yang berkomunikasi dengan *node* lain melalui sebuah *node* tertentu atau *node relay*. Jaringan *multihop* diterapkan pada komunikasi antar *node* yang tidak memungkinkan secara langsung karena jarak atau kendala komunikasi tertentu.

2.2.2 Arduino Nano

Arduino Nano adalah Mikrokontroler berukuran kecil yang mendukung penggunaan breadboard. Diciptakan dengan basis Mikrokontroler ATmega328 untuk versi 3.x. Arduino Nano tidak menyertakan colokan DC berjenis Barrel Jack, melainkan dihubungkan ke komputer menggunakan port USB Mini-B (Arduino,

2017). Bentuk dari arduino nano dapat dilihat pada Gambar 2.2. dan spesifikasinya dapat dilihat pada Tabel 2.1.



Gambar 2.2 Board Arduino Nano

Sumber: Arduino (2017)

Tabel 2.1 Spesifikasi Arduino Nano

Mikrokontroler	ATmega328
Tegangan Operasi	5 V
Memori Flash	32 KB (2 KB digunakan untuk bootloader)
Pin IN Analog	8
EEPROM	1 KB
Kecepatan Jam	16 MHz
Tegangan Input	7-12 V
Arus DC per I/O Pin	40 mA
Konsumsi Daya	19 mA
I/O Pin Digital	22 (6 pin diantaranya adalah PWM)

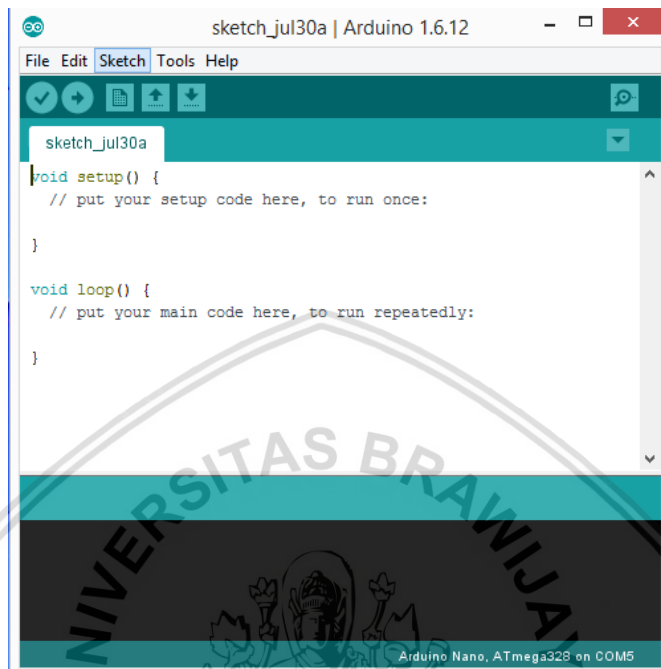
Sumber: Arduino (2017)

Semua pin pada Arduino Nano beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor pull-up internal sebesar 20-50 KOhm. Selain itu beberapa pin memiliki fungsi khusus, yaitu:

1. Serial : 0 (RX) dan 1 (TX). RX digunakan untuk menerima dan TX digunakan untuk mengirimkan data.
2. Pin 2 dan 3 dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah.
3. Pin 3, 5, 6, 9, 10, dan 11. Menyediakan output PWM 8-bit dengan fungsi `analogWrite()`.
4. SPI : Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin ini mendukung komunikasi SPI.
5. Pin 13. LED terhubung ke pin digital 13.
6. Pin A4 (SDA) dan pin A5 (SCL). Yang mendukung komunikasi I2C yang menggunakan perpustakaan Wire.
7. AREF : Referensi tegangan untuk input analog.
8. RESET : Digunakan untuk me-reset (menghidupkan ulang) Mikrokontroler.

2.2.3 Arduino IDE

Arduino IDE merupakan sebuah software yang berperan sebagai media menulis program, meng-compile menjadi kode biner dan meng-upload kedalam *memory* Mikrokontroller. Interface Arduino IDE dapat dilihat pada Gambar 2.3.



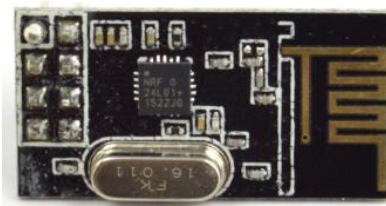
Gambar 2.3 Interface Arduino IDE

Arduino IDE terdiri dari :

1. *Editor* program, sebuah window memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. *Compile*, sebuah modul yang mengubah kode program (bahasa *processing*) menjadi biner. Bagaimanapun sebuah Mikrokontroller tidak akan bisa memahami bahasa *processing*.
3. *Uploader*, sebuah modul yang memuat kode biner dari komputer ke dalam memori di dalam board Arduino

2.2.4 Modul nRF24L01

nRF24L01 adalah modul komunikasi dengan pita frekuensi 2.4GHz ISM (*Industrial, Cientific and Medical*). Radio chip ini dikonfigurasi dan dioperasikan melalui antarmuka serial. nRF24L01 memiliki solusi terkait daya, berupa daya ultra rendah yang memungkinkan daya tahan baterai bisa sampai berbulan-bulan, konsumsi daya hanya 9.0 mA pada output –6 dBm dan 12.3 mA dalam mode RX (Nordic Semiconductor, 2007). Modul nRF24L01 dapat dilihat pada Gambar 2.4 dan untuk modul pin pada nRF24L01 dapat dilihat pada Tabel 2.2.



Gambar 2.4 Modul nRF24L01

Sumber: <https://www.robotshop.com/en/transceiver-nRF24L01-module.html>

Tabel 2.2 Fungsi Pin Pada Modul nRF24L01

Pin	Nama	Fungsi Pin	Deskripsi
1	GND	Power Source	Sumber tegangan (+1.9V - +3.6V DC)
2	VCC	Power Source	Ground (0V)
3	CE	Input Digital	Untuk mengaktifkan RX dan TX
4	CSN	Input Digital	Untuk memilih mode SPI
5	SCK	Input Digital	Waktu SPI
6	MOSI	Input Digital	Data input SPI
7	MISO	Output Digital	Data output SPI
8	IRQ	Output Digital	Pin interrupt maskable

Sumber: Nordic Semiconductor (2007)

2.2.4.1 Fitur-fitur Modul nRF24L01

nRF24L01 memiliki banyak fitur di dalamnya, dengan mengetahui fitur-fitur yang dimiliki modul nRF24L01 diharapkan dapat membantu dalam penelitian ini. Fitur-fitur modul nRF24L01 yaitu:

1. Radio

- Menggunakan frekuensi 2.4Ghz global ISM *band*
- 126 RF *channel*
- GFSK modulasi
- Data *rate up* 2Mbps
- 1 MHz jarak *non-overlapping channel* pada 1Mbps
- 2 MHz jarak *non-overlapping channel* pada 2Mbps

2. Transmitter

- Output daya: 0, -6, -12 atau -18dBm
- 11.3mA pada 0dBm *output* daya

3. Receiver

- *Filter channels* yang terintegrasi
- 12.3mA pada 2Mbps
- -82dBm sensitif pada 2Mbps
- -85dBm sensitif pada 1Mbps
- -94dBm sensitif pada 250 Kbps

4. RF Synthesizer

- *Synthesizer* yang terintegrasi

- Menerima biaya rendah $\pm 60\text{ppm}$ 16MHz

5. Enhance ShockBurst

- Panjang *payload* 1 sampai 32 byte
- Penanganan paket secara otomatis
- Penanganan transaksi paket secara otomatis

6. Power Management

- regulator tegangan terintegrasi
- catu daya 1.9 ke 3.6V
- 26 μA mode *standby*, pasokan daya pada mode power rendah 900nA

7. Host Interface

- 4-pin SPI
- Max 8Mbps
- *input* bertoleransi 5V

2.2.5 RF24Network

RF24Network menggunakan metode sederhana kompresi data untuk menyimpan alamat hanya menggunakan 2 byte, dalam format yang dirancang untuk mewakili topologi jaringan dengan cara yang intuitif (Doxxygen, 2016).

2.2.5.1 Radio Komunikasi dan Topologi

Radio dalam jaringan ini dihubungkan oleh alamat yang ditugaskan ke *pipe*. Setiap radio dapat mendengarkan 6 alamat pada 6 *pipe*, oleh karena itu setiap radio memiliki *pipe* induk dan 5 *pipe* anak, yang digunakan untuk membentuk struktur *tree*. Struktur topologi dapat dilihat pada Gambar 2.5.

RF24Network Topology:

Lines — = Data Connections
Addresses are assigned in groups of 5:
In the following chart: "n" = 01 to 05

Example:
Node 00 transmits a message to Node 02125. The message will pass through nodes 05, 025, 0125, then to 2125.

Example:
Node 00(Master) wants to send a message to nodes 011, 021, and 051. The messages pass through node 01, but the message to node 051 fails. Node 051 has a dead battery.

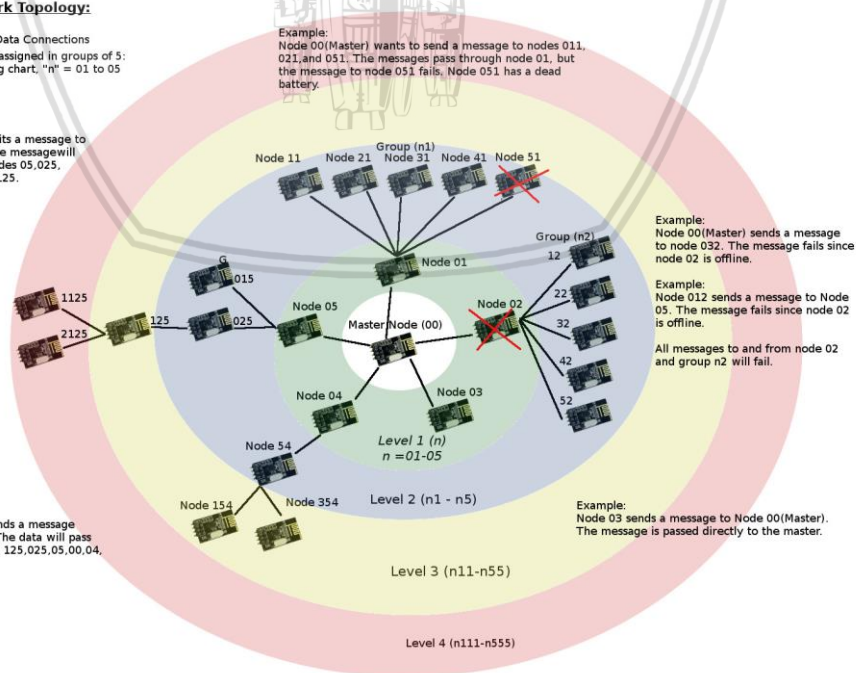
Example:
Node 00(Master) sends a message to node 032. The message fails since node 02 is offline.

Example:
Node 012 sends a message to Node 05. The message fails since node 02 is offline.

All messages to and from node 02 and group n2 will fail.

Example:
Node 1125 sends a message to node 054. The data will pass through nodes 125, 025, 05, 00, 04, then to 054.

Example:
Node 03 sends a message to Node 00(Master). The message is passed directly to the master.



Gambar 2.5 Topologi RF24Network

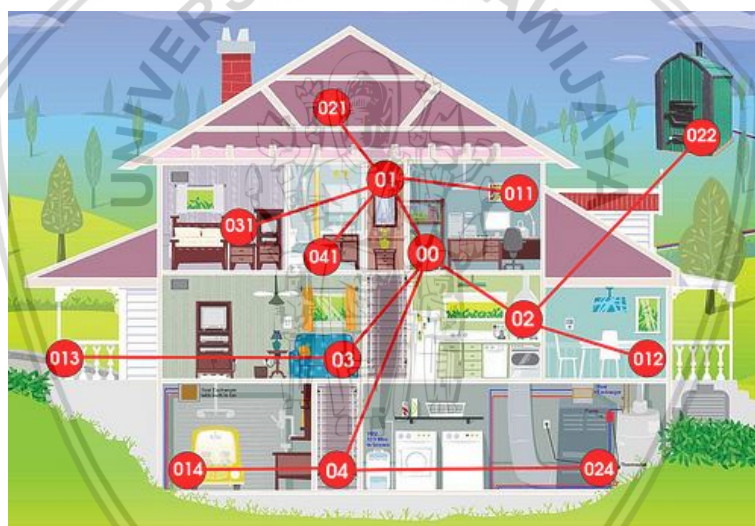
Sumber : <https://tmrh20.github.io/RF24Network/Tuning.html>

Ketika transmisi berlangsung dari satu modul radio ke yang lain, radio *receiver* akan berkomunikasi kembali ke *transmitter* dengan paket pengakuan (ACK), untuk menunjukkan keberhasilan. Jika *node transmitter* tidak menerima ACK, radio secara otomatis terlibat dalam serangkaian retries waktu, pada interval yang ditetapkan. Radio menggunakan teknik seperti pengalamatan dan penomoran muatan untuk mengelola, tetapi semuanya dilakukan secara otomatis oleh chip nRF, yang tidak terlihat oleh pengguna.

Ketika bekerja melalui jaringan radio, beberapa teknik otomatis ini benar-benar dapat menghambat transmisi data hingga taraf tertentu. Mencoba kembali *payload* yang gagal berulang kali di jaringan radio dapat menghambat komunikasi untuk *node* di dekatnya, atau mengurangi *throughput* dan kesalahan pada *node routing*.

2.2.5.2 Pengalamatan

Untuk pengalamatan RF24Network menetapkan *nodenya* secara bertingkat menggunakan bilangan *octal*. Setiap radio memiliki *pipe* induk dan 5 *pipe* anak. Contoh pemetaan *node* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Pengalamatan Pada RF24Network

Sumber: www.maniacbug.wordpress.com

2.2.5.3 Routing

Routing lalu lintas ditangani secara tidak terlihat menggunakan *network layer*. Pengalamatan jaringan sesuai dengan tata letak *physical network*, *node* akan mengarahkan *traffic* secara otomatis sesuai dengan kebutuhan. Pengguna cukup membangun header yang berisi alamat tujuan yang tepat, dan jaringan akan meneruskannya ke *node* tujuan. Ketika *payload* dikirim ke *node* lain, mereka harus dirutekan. *Routing* dikelola menggunakan kombinasi permintaan ACK yang dibangun, memungkinkan semua *node routing* untuk meneruskan data dengan sangat cepat, dengan hanya *node routing* akhir yang mengkonfirmasi pengiriman dan mengirim kembali pengakuan ACK.

2.2.6 DHT11

DHT11 merupakan sensor dengan kalibrasi sinyal digital yang mampu memberikan informasi suhu dan kelembaban. Sensor ini tergolong komponen yang memiliki tingkat stabilitas yang sangat baik, dengan kemampuan Mikrokontroler 8 bit. Kualitas dari sensor ini bisa dikatakan terbaik, memiliki respon pembacaan yang cepat, dan kemampuan anti interferensi serta dengan harga yang sangat terjangkau. Sensor DHT11 dapat dilihat pada Gambar 2.7.



Gambar 2.7 Sensor DHT11

Sumber: <https://www.ebay.co.uk/itm/DHT11-Temperature-and-Humidity-Sensor-Module-for-Arduino-/291654837050>

Spesifikasi DHT11 adalah sebagai berikut:

1. Sumber Tegangan : +5 V
2. Kisaran Suhu : 0-50 °C kesalahan sekitar ± 2 °C
3. Kisaran Kelembaban : 20-90% RH kesalahan sekitar ± 5 % RH
4. Interface : Digital

2.2.7 Pemrograman Python

Pemrograman Python adalah bahasa pemrograman tinggi yang dapat melakukan eksekusi sejumlah intruksi multiguna secara langsung dengan metode orientasi objek (*Object Oriented Programming*) serta menggunakan semantik dinamis untuk memberikan tingkat keterbacaan *syntax*. Python bisa digunakan untuk berbagai keperluan pengembangan perangkat lunak dan bisa berjalan di berbagai platform sistem operasi seperti Linux, Windows, Mac OS X, Java Virtual Machine, OS/2, Amiga, Palm, Symbian.

PySerial adalah library/modul python yang dibuat untuk memudahkan dalam komunikasi data serial. Pengguna cukup menambahkan (menginstall) library dan memanggilnya kedalam sebuah *syntax* untuk keperluan tertentu.

2.2.8 Pengukuran Jaringan Nirkabel

Menjelaskan parameter-parameter yang digunakan untuk mengukur kinerja jaringan. Dimana parameter yang digunakan untuk pengukuran kinerja antara lain adalah:

2.2.8.1 Delay

Delay merupakan waktu yang dibutuhkan dari pengiriman paket data yang dikirim sampai paket data yang diterima. *Delay* dapat dipengaruhi oleh jarak,

media fisik, atau juga waktu proses yang lama (Iqbal, 2013). Kategori *delay* dapat dilihat pada Tabel 2.3.

Tabel 2.3 Kategori *Delay*

Kategori Delay	Besar Delay
Sangat Bagus	<150 ms
Bagus	150 s/d 300 ms
Sedang	300 s/d 450 ms
Jelek	>450 ms

$$Delay = \frac{(Waktu\ penerimaan\ paket - Waktu\ pengiriman\ paket)}{Waktu\ penerimaan\ paket} \quad (2.1)$$

$$Rata-rata\ delay = \frac{Total\ delay}{Jumlah\ paket\ dikirim} \quad (2.2)$$

2.2.8.2 Packet loss

Packet loss merupakan parameter uji yang menggambarkan kondisi dari jumlah paket yang hilang selama proses transmisi dari *transmitter* ke *receiver*. *Packet loss* terjadi ketika satu atau lebih paket data yang melewati suatu jaringan gagal mencapai tujuannya. Kategori degradasi *packet loss* dapat dilihat pada Tabel 2.4.

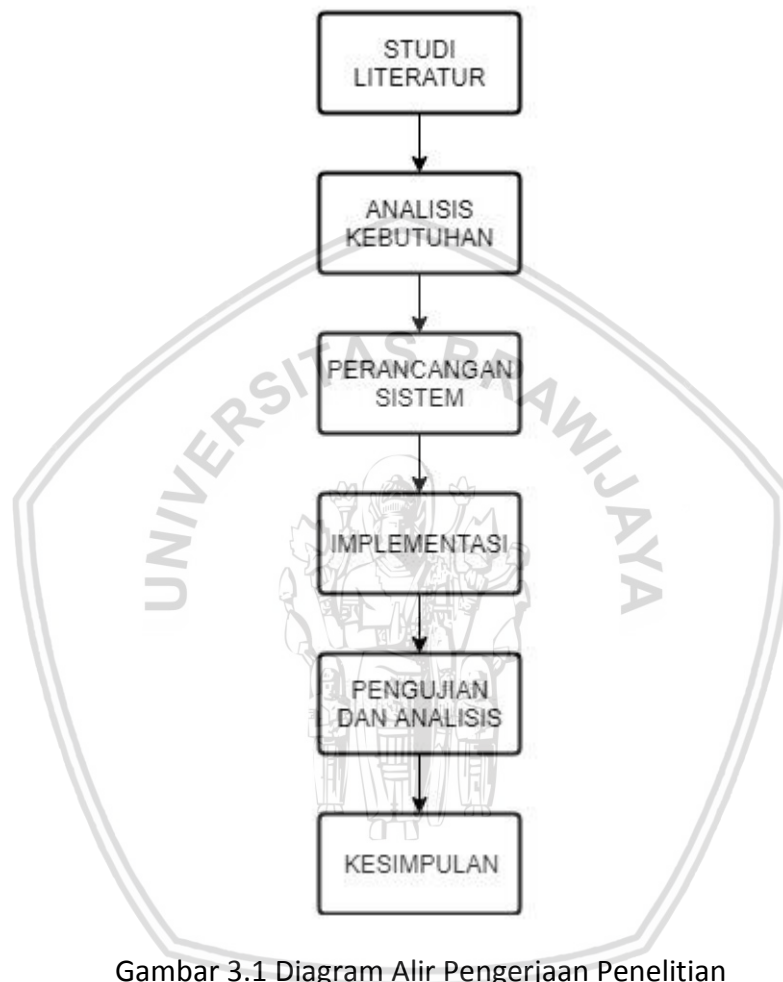
Tabel 2.4 Kategori Degradasi *Packet loss*

Kategori Degradasi	Paket Loss	Indeks
Sangat Bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Jelek	25%	1

$$Packet\ loss = \frac{(Jumlah\ paket\ dikirim - Jumlah\ paket\ diterima)}{Jumlah\ paket\ dikirim} \times 100\% \quad (2.3)$$

BAB 3 METODOLOGI

Menjelaskan mengenai tahapan yang dilakukan untuk penyusunan penelitian yang meliputi : studi literatur, analisis kebutuhan sistem, perancangan sistem, implementasi, pengujian dan analisa, kesimpulan. Untuk diagram alir pengerjaan penelitian ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Pengerjaan Penelitian

3.1 Studi Literatur.

Studi literatur dilakukan untuk memahami konsep yang berhubungan dengan penelitian ini. Studi literatur ini digunakan untuk menunjang pemahaman yang dibutuhkan untuk mengerjakan penulisan laporan dan pelaksanaan penelitian. Referensi utama yang diperlukan untuk menunjang penulisan ini adalah :

1. Jaringan Sensor Nirkabel (JSN)
2. Arduino Nano
3. nRF24L01
4. RF24Network
5. DHT11

6. Python
7. Parameter Pengukuran Jaringan

3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dibutuhkan untuk mendapatkan gambaran tentang sistem yang akan dirancang, analisa kebutuhan sistem mencakup kebutuhan fungsional dan non-fungsional.

Kebutuhan fungsional pada sistem ini antara lain :

1. *Node transmitter* dapat mengakuisisi sensor suhu dan kelembaban.
2. *Node transmitter* dapat mengirimkan data sensor ke *node reciver*.
3. *Node receiver* dapat menerima data sensor yang dikirim dari *node transmitter*.
4. Jaringan sensor nirkabel dapat berkomunikasi menggunakan mekanisme *multi-hop*.

Analisis kebutuhan non-fungsional mencakup kebutuhan yang terkait dengan perangkat keras dan perangkat lunak. Kebutuhan non-fungsional pada sistem ini antara lain :

1. Kebutuhan Perangkat Keras :
 - a. 2 buah PC/Laptop.
 - b. 5 buah Arduino Nano sebagai Mikrokontroler.
 - c. 4 buah nRF24L01 sebagai *transmitter* dan 1 buah nRF24L01 sebagai *receiver*.
 - d. 3 buah power bank sebagai catu daya.
 - e. 5 buah Kabel USB.
 - f. 4 buah DHT11 untuk sensor suhu dan kelembaban.
 - g. kabel jumper.
 - h. 5 buah *capasitor* 10uf.
2. Kebutuhan Perangkat Lunak :
 - a. Arduino IDE.
 - b. Python

3.3 Perancangan Sistem

Perancangan sistem dibutuhkan untuk dapat mengerti rancangan yang akan dibuat. Pada perancangan sistem terdapat 2 tahapan, dapat dilihat pada diagram blok perancangan sistem pada Gambar 3.2.



Gambar 3.2 Diagram blok perancangan sistem

3.3.1 Perancangan Perangkat Keras

Perancangan perangkat keras digunakan untuk merancang node-node yang akan dibuat, baik *transmitter* maupun *receiver*. Adapaun kebutuhan perangkat keras dalam pembuatan sistem di antaranya:

- Arduino Nano
- nRF24L01
- Komputer(PC)/Laptop
- Kabel USB
- Kabel Jumper
- Capacitor
- DHT11
- Powerbank

3.3.2 Perancangan Perangkat Lunak

Perancangan Perangkat Lunak digunakan untuk merancang source code dan mekanisme bagaimana cara rangkaian perangkat keras dapat bekerja sesuai dengan kebutuhan. Adapaun perangkat lunak yang dibutuhkan untuk mendukung pembuatan sistem adalah :

1. Arduino IDE dengan Spesifikasi sebagai berikut :

Sistem Operasi	Windows 8.1 64 Bit
Bahasa Pemrograman	C (Arduino)
Tools Pemrograman	Arduino IDE versi 1.6.12
Kegunaan	Untuk merancang source code dan mengupload source code ke board Arduino

2. Python 2.7.3

Sistem Operasi	Windows 8.1 64 Bit
Bahasa Pemrograman	Python
Kegunaan	Mendapatkan waktu PC/laptop untuk menghitung <i>delay</i> .

3.4 Implementasi

Implementasi pada penelitian ini mengacu pada perancangan yang telah dibuat pada tahap sebelumnya yang akan menerapkan semua yang telah dirancang dan diimplementasikan. Tahap ini merupakan tahap yang sangat menentukan berhasil tidaknya rancangan yang dibuat. Implementasi sistem dibagi menjadi 2 yaitu implementasi perangkat keras dan implementasi perangkat lunak.

3.4.1 Implementasi Perangkat Keras

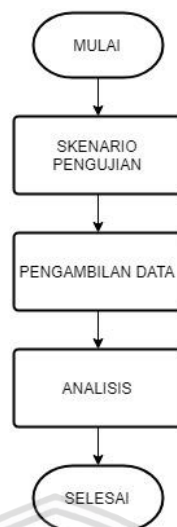
Setelah dilakukan perancangan perangkat keras juga dibutuhkan implementasi perangkat keras, dimana perancangan yang sudah dibuat diaplikasikan sehingga perangkat keras satu dengan yang lainnya dapat terhubung sehingga dapat terbentuk *node* sensor. *Node* sensor terdiri dari Arduino Nano, nRF24L01, dan DHT11.

3.4.2 Implementasi Perangkat Lunak

Setelah dilakukan perancangan perangkat keras juga dibutuhkan implementasi perangkat lunak, perancangan yang sudah dibuat diaplikasikan agar setiap perangkat keras yang digabungkan mampu menjalankan fungsinya.

3.5 Pengujian dan Analisis

Pada tahap ini skenario pengujian dilakukan untuk melihat bagaimana kinerja dari hasil sistem yang dibuat. Analisa yang dilakukan pada penelitian ini mengacu pada *packet loss* dan *delay*. Tahapan pengujian dan analisis digambarkan dalam bentuk diagram alir seperti Gambar 3.3.



Gambar 3.3 Diagram Alir Pengujian dan Analisis

3.5.1 Pengujian

Terdapat beberapa pengujian yang akan dilakukan pada penelitian ini dengan skenario pengujian yaitu :

1. Pengujian perangkat keras, untuk mengetahui apakah *node* sensor berjalan dengan baik atau tidak.
2. Pengujian perangkat lunak, untuk mengetahui apakah perangkat keras sudah berjalan sesuai dengan fungsinya.
3. Pengujian mekanisme *single-hop* pada ruang terbuka dan ruang tertutup dengan parameter uji *packet loss* dan *delay*.
4. Pengujian mekanisme *multi-hop* pada ruang terbuka dan ruang tertutup dengan parameter uji *packet loss* dan *delay*.

3.5.2 Analisis

Dari pengujian yang telah dilakukan, data dari masing-masing tahap pengujian akan dianalisis untuk mengetahui kinerja dari implementasi sistem dan untuk menarik kesimpulan dari penelitian apakah sistem yang dibuat sudah sesuai dengan tujuan penulisan.

3.6 Kesimpulan dan Saran

Pengambilan kesimpulan dan saran dilakukan setelah semua tahapan selesai untuk mengetahui hasil penelitian dari pengujian sistem. Kesimpulan ini diambil dari hasil akhir pada tahap pengujian dan analisis dari perancangan sistem yang dibangun. Sedangkan saran adalah tahap terakhir penulisan yang bertujuan untuk memperbaiki kesalahan yang terjadi dan dapat disempurnakan lagi oleh penelitian berikutnya serta untuk memberikan pertimbangan pada pengembangan perancangan sistem selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Bab ini memberikan penjelasan mengenai kebutuhan-kebutuhan yang harus dipenuhi baik itu kebutuhan fungsional maupun kebutuhan non fungsional agar nantinya sistem dapat berjalan sesuai dengan penelitian yang dilakukan.

4.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan fungsionalitas dari sistem yang akan diimplementasikan. Kebutuhan fungsional dari sistem yang akan dibuat meliputi :

1. DHT11 dapat mengakuisisi suhu dan kelembaban ruangan.

Diharapkan DHT11 dapat mengakuisisi suhu dan kelembaban untuk mengetahui kondisi ruangan, yang nantinya data suhu dan kelembaban akan dikirimkan ke *node receiver*.

2. *Node transmitter* dapat mengirimkan data sensor ke *node receiver*.

Node transmitter diharapkan mampu berkomunikasi dengan *node receiver*, bisa mengirimkan data sensor kepada *node receiver* menggunakan modul nRF24L01 berbasis protokol RF24Network.

3. *Node receiver* dapat menerima data sensor dari *node transmitter*.

Node receiver diharapkan bisa menerima data yang dikirimkan oleh *node transmitter* menggunakan modul nRF24L01 berbasis protokol RF24Network.

4. Jaringan sensor nirkabel mampu berkomunikasi menggunakan mekanisme *multi-hop*.

Diharapkan antar *node* bisa berkomunikasi dengan mekanisme *multi-hop* menggunakan protokol RF24Network.

4.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan opsional agar sistem dapat berjalan lebih baik. Kebutuhan non fungsional mencakup kebutuhan perangkat keras dan kebutuhan perangkat lunak.

4.2.1 Kebutuhan Perangkat Keras

Bagian ini menjelaskan terkait akan kebutuhan perangkat keras pada sistem ini, yang meliputi sekumpulan komponen-komponen elektronika yang nantinya akan membentuk suatu sistem baru dengan fungsionalitas tertentu. Berikut adalah komponen yang dibutuhkan dalam implementasi sistem yang akan dibuat.

1. Arduino Nano

Mikrokontroler ATmega328 (Arduino Nano) digunakan untuk menjalankan kode program dan untuk menghubungkan dengan media komunikasi nRF24L01 menggunakan SPI serta pembacaan data sensor.

2. Sensor DHT11

Sensor DHT11 berfungsi sebagai media akuisisi data pada Jaringan Sensor Nirkabel. DHT11 adalah sensor suhu dan kelembaban, dia memiliki keluaran sinyal digital yang dikalibrasi dengan sensor suhu dan kelembaban yang kompleks.

3. Modul nRF24L01

Modul nRF24L01 merupakan modul komunikasi yang berfungsi untuk mengirimkan data antar *node*. Modul ini menggunakan sinyal radio dengan frekuensi 2,4 GHz. Dibutuhkan 2 modul untuk bisa berkomunikasi antar *node*, dimana 1 *node* sebagai *transmitter* dan *node* lainnya sebagai *receiver*.

4.2.2 Kebutuhan Perangkat Lunak

Bagian ini menjelaskan terkait akan kebutuhan perangkat lunak pada sistem ini, perangkat lunak yang terkait merupakan software yang nantinya akan digunakan untuk membangun sistem agar dapat beroperasi sesuai dengan fungsionalitasnya. Berikut adalah perangkat lunak yang dibutuhkan untuk implementasi sistem yang akan dibuat.

1. Arduino IDE

Integrated Development Environment (IDE) adalah sebuah software yang berperan sebagai media menulis program, meng-compile menjadi kode biner dan meng-upload kedalam *memory* mikrokontroler. Software ini akan sangat dibutuhkan untuk mengisi dan mengedit kode program pada Arduino Nano.

2. Python

Python merupakan bahasa pemrograman yang bisa digunakan untuk multiguna. Dan nantinya bahasa pemrograman ini akan digunakan untuk mengambil waktu PC/Laptop untuk mengetahui waktu pengiriman dan penerimaan data antar *node* menggunakan komunikasi serial untuk menangkap aktifitas yang dilakukan oleh Arduino Nano.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

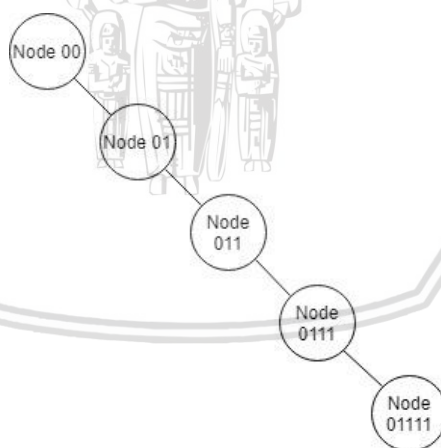
Bab ini memberikan penjelasan terkait perancangan dan proses implementasi yang dilakukan dalam pembuatan sistem, yang meliputi perancangan topologi jaringan yang akan digunakan, perancangan sistem menggunakan diagram blok, skematik perancangan perangkat keras, diagram alir perangkat lunak, serta implelementasi perangkat keras dan impelementasi perangkat lunak.

5.1 Perancangan Sistem

Bagian ini menjelaskan mengenai proses perancangan sistem terkait perancangan topologi jaringan, perancangan perangkat keras yang meliputi diagram blok sistem dan skematik perancangan perangkat keras. Serta perancangan perangkat lunak menggunakan diagram alir yang akan dijelaskan menggunakan flowchart.

5.1.1 Topologi Jaringan

Agar nantinya sistem dapat berjalan sesuai dengan penelitian, dilakukan perancangan topologi yang akan digunakan. RF24Network untuk pengalamatannya menggunakan bilangan octal. Dan pada penelitian ini implemententasi yang dilakukan menggunakan 5 Arduino Nano. Didapatkan perancangan topologi yang nantinya akan diterapkan pada implementasi sistem. Perancangan topologi yang digunakan dapat dilihat pada Gambar 5.1.



Gambar 5.1 Topologi Jaringan

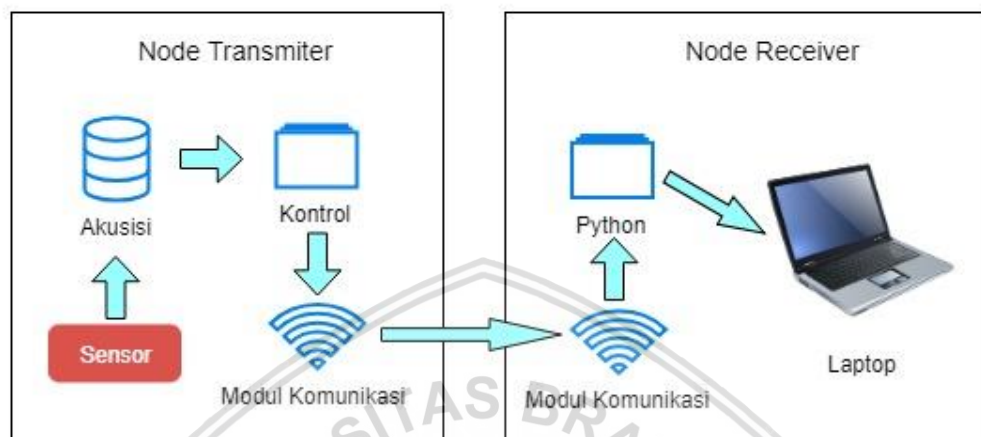
5.1.2 Perancangan Perangkat Keras

Pada perancangan perangkat keras mempunyai 2 metode pendekatan, yaitu menggunakan diagram blok sistem dan skematik perancangan perangkat keras.

5.1.2.1 Diagram Blok Sistem

Terdapat 2 perancangan yang akan diimplementasikan pada sistem ini, yang meliputi *node transmitter* dan *node receiver*. Kedua perangkat tersebut memiliki

fungsi yang berbeda, perangkat keras *node transmitter* memiliki fungsi untuk melakukan akuisisi data sensor dan mengirimkan data sensor kepada *node receiver* sedangkan perangkat keras *node receiver* memiliki fungsi untuk menerima data sensor yang dikirim oleh *node transmitter*. Diagram blok sistem *node transmitter* dan *node receiver* dapat dilihat pada Gambar 5.2.



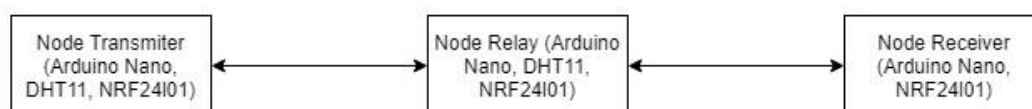
Gambar 5.2 Diagram Blok Sistem

Dan untuk diagram blok mekanisme *single-hop* dan *multi-hop* dapat dilihat pada Gambar 5.3 dan 5.4.



Gambar 5.3 Diagram Blok Komunikasi *Single-hop*

Gambar 5.3 menggambarkan mekanisme *single-hop*. *Node transmitter* mengirimkan data sensor kepada *node receiver*, dan *node receiver* menerima data sensor yang dikirimkan oleh *node transmitter* berupa data suhu dan kelembaban.



Gambar 5.4 Diagram Blok *Multi-hop*

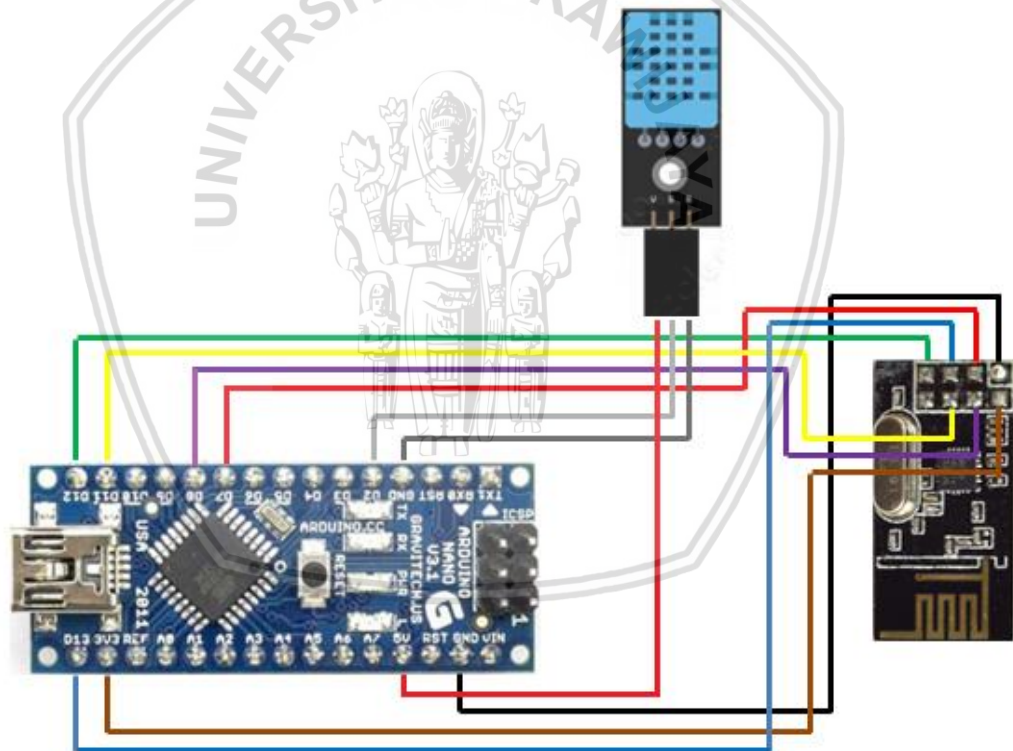
Gambar 5.4 menggambarkan *multi-hop*. Agar *node transmitter* bisa mengirimkan data sensor ke *node receiver* dibutuhkan *node relay* untuk meneruskan data. Jika *node relay* dalam keadaan *offline* maka data yang dikirimkan dari *node transmitter* menuju *node receiver* tidak akan terkirim. Data yang diterima *node receiver* merupakan data dari *node relay* dan *node transmitter*.

5.1.2.2 Skematik Perancangan Perangkat Keras

Tahap perancangan perangkat keras mempunyai 2 sistematika perancangan sistem, yaitu meliputi perancangan perangkat keras *transmitter* dan perancangan perangkat keras receiver.

1. Perancangan Perangkat Keras *Transmitter*

Perancangan perangkat keras pada *node transmitter* terdiri dari Mikrokontroler, sensor, dan modul wireless. *Node transmitter* menggunakan ATmega328 (Arduino Nano) sebagai Mikrokontroler. Catu daya yang digunakan oleh *node transmitter* menggunakan laptop/powerbank dan kabel USB sebagai penghubung untuk catu daya. Untuk rangkaian sensor pada *node transmitter* menggunakan DHT11 untuk mengukur suhu dan kelembaban lingkungan area di sekitar *node*. Dan modul wireless yang digunakan adalah nRF24L01 yang berfungsi untuk mengirimkan data ke *node receiver*. Rangkaian dari *node sensor transmitter* dapat dilihat pada Gambar 5.5 dan untuk koneksi pin *node transmitter* dapat dilihat pada Tabel 5.1 dan 5.2.



Gambar 5.5 *Node Transmitter*

Tabel 5.1 Koneksi pin Arduino Nano dengan nRF24L01

Pin Arduino Nano	Pin NRF24L01	Warna Kabel
VCC	VCC	Brown
GND	GND	Black
D13	SCK	Blue
D12	MISO	Green
D11	MOSI	Yellow
D7	CE	Red
D8	CSN	Purple

Dilihat pada Gambar 5.5 dan Tabel 5.1 untuk menghubungkan Arduino Nano dengan nRF24L01.

1. Pin VCC pada nRF24L01 dihubungkan dengan pin VCC 3V pada Arduino Nano.
2. Pin GND pada nRF24L01 dihubungkan dengan pin GND pada Arduino Nano
3. Pin SCK pada nRF24L01 dihubungkan dengan pin D13 pada Arduino Nano.
4. Pin MISO pada nRF24L01 dihubungkan dengan pin D12 pada Arduino Nano.
5. Pin MOSI pada nRF24L01 dihubungkan dengan pin D11 pada Arduino Nano.
6. Pin CE pada nRF24L01 dihubungkan dengan pin D7 pada Arduino Aano.
7. Pin CSN pada nRF24L01 dihubungkan dengan pin D8 pada Arduino Nano.

Tabel 5.2 Koneksi Pin Arduino Nano dengan DHT11

Pin Arduino Nano	Pin DHT11	Warna Kabel
VCC	VCC	Red
GND	GND	Black
D2	DATA	Grey

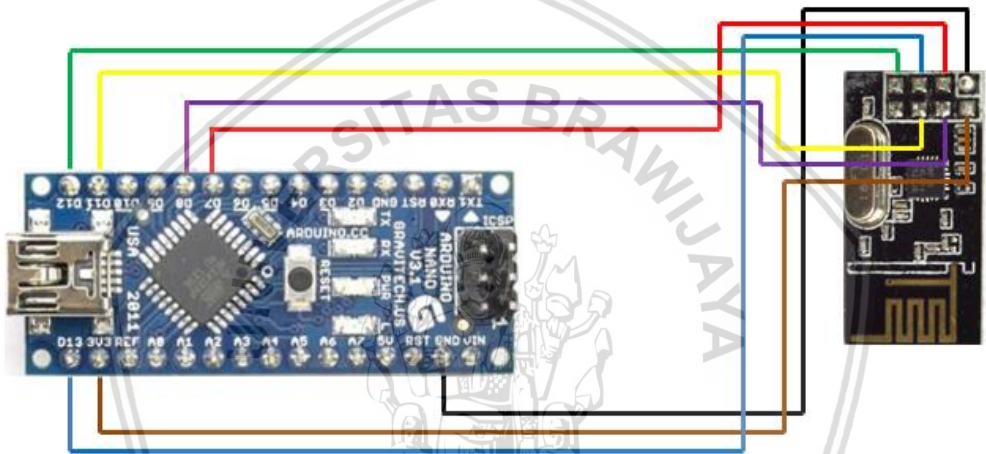
Dan dilihat pada Gambar 5.4 dan tabel 5.2 untuk menghubungkan Arduino Nano dengan sensor DHT11.

1. Pin VCC pada DHT11 dihubungkan dengan pin VCC 5V pada Arduino Nano.

- 2. Pin GND pada DHT11 dihubungkan dengan pin GND pada Arduino Nano.
- 3. Pin Data pada DHT11 dihubungkan dengan pin D2 pada Arduino Nano.

2. Perancangan Perangkat Keras *Receiver*

Perancangan perangkat keras pada *node receiver* terdiri dari Mikrokontroler dan modul wireless. *Node receiver* menggunakan Mikrokontroler yang sama yaitu Mikrokontroler Atmega328 (Arduino Nano). Catu daya yang digunakan oleh *node receiver* menggunakan laptop, kabel USB sebagai penghubung antara Mikrokontroler dengan laptop. Modul wireless yang digunakan *node receiver* juga sama dengan *node transmitter* yaitu modul nRF24L01 yang berfungsi untuk menerima data sensor dari *node transmitter*. Rangkaian dari *node sensor transmitter* dapat dilihat pada Gambar 5.6 dan untuk koneksi pin *node transmitter* dapat dilihat pada tabel 5.3.



Gambar 5.6 *Node Receiver*

Tabel 5.3 Koneksi Pin Arduino Nano dengan nRF24L01

Pin Arduino Nano	Pin NRF24L01	Warna Kabel
VCC	VCC	Brown
GND	GND	Black
D13	SCK	Blue
D12	MISO	Green
D11	MOSI	Yellow
D7	CE	Red
D8	CSN	Purple

Dilihat pada Gambar 5.6 dan Tabel 5.3 untuk menghubungkan Arduino Nano dengan nRF24L01.

1. Pin VCC pada nRF24L01 dihubungkan dengan pin VCC 3V pada Arduino Nano.
2. Pin GND pada nRF24L01 dihubungkan dengan pin GND pada Arduino Nano.
3. Pin SCK pada nRF24L01 dihubungkan dengan pin D13 pada Arduino nano.
4. Pin MISO pada nRF24L01 dihubungkan dengan pin D12 pada Arduino Nano.
5. Pin MOSI pada nRF24L01 dihubungkan dengan pin D11 pada Arduino Nano.
6. Pin CE pada nRF24L01 dihubungkan dengan pin D7 pada Arduino Nano.
7. Pin CSN pada nRF24L01 dihubungkan dengan pin D8 pada Arduino Nano.

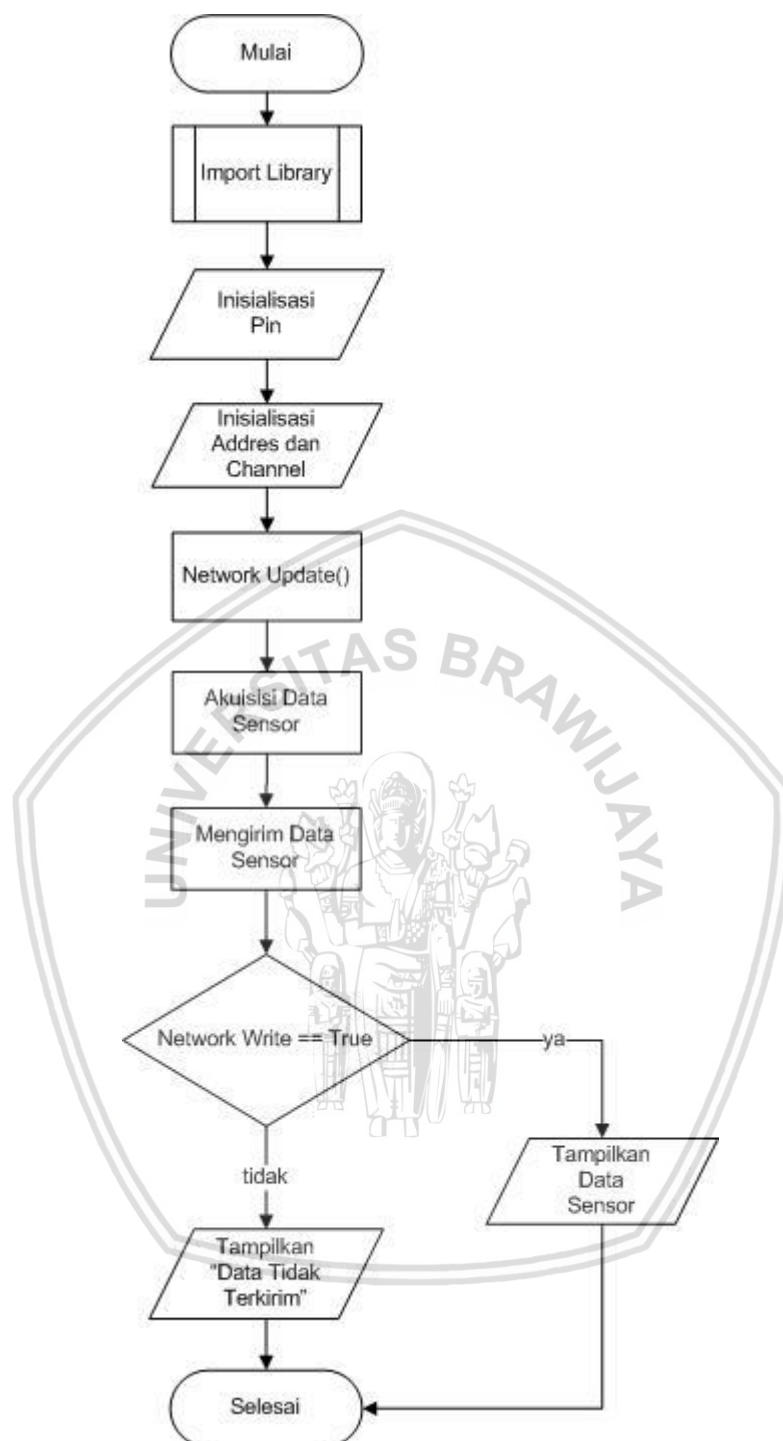
5.1.3 Perancangan Perangkat Lunak

Sub bab ini menjelaskan mengenai perancangan perangkat lunak pada sistem yang akan dibuat. Perancangan pemodelan pada sistem yang akan diimplementasikan akan diuraikan pada sub ini.

5.1.3.1 Perancangan Perangkat Lunak *Transmitter*

Perancangan perangkat lunak pada *node transmitter* menggunakan bahasa C yang diprogram menggunakan Arduino IDE. Pada perancangan ini diharapkan *node transmitter* dapat mengakuisisi data sensor dan mengirimkan data sensor kepada *node receiver*.

Agar nantinya *node* sensor dapat bekerja sesuai dengan fungsinya, terlebih dahulu dilakukan import *library* agar perangkat keras dan algoritme dapat dikenali dan berjalan dengan baik. Setelah import *library* dilakukan inisialisasi pin yang digunakan oleh nRF24L01 dan DHT11 kemudian diikuti dengan pemberian alamat *node* dan *channel*. Protokol RF24Network menggunakan bilangan octal untuk pengalamatannya. Pengalamatan *node* secara bertingkat sesuai dengan turunan dari *node master*. Dan *channel* yang digunakan harus sama dengan yang digunakan oleh *node receiver* yaitu *channel* 90 agar nantinya data dapat terkirim ke *node receiver*. Setelah pemberian alamat dan *channel* selanjutnya *node transmitter* mengecek apakah jaringan yang akan digunakan tersedia melalui perintah *network.update()*. Kemudian dilakukan proses pembacaan data sensor, data sensor yang diakuisisi merupakan data sensor suhu dan kelembaban di sekitar *node* yang didapatkan dari sensor DHT11. Pada saat proses pengiriman data akan ada kondisi untuk melihat apakah *node receiver* dalam keadaan hidup atau mati. Jika *node receiver* ready data sensor akan ditampilkan di layar komputer. Namun jika *node receiver* dalam keadaan offline data sensor tidak akan ditampilkan di layar komputer melainkan hanya menampilkan pesan teks berupa "Data Tidak Terkirim". Untuk perancangan perangkat lunak *transmitter* dapat dilihat pada Gambar 5.7.

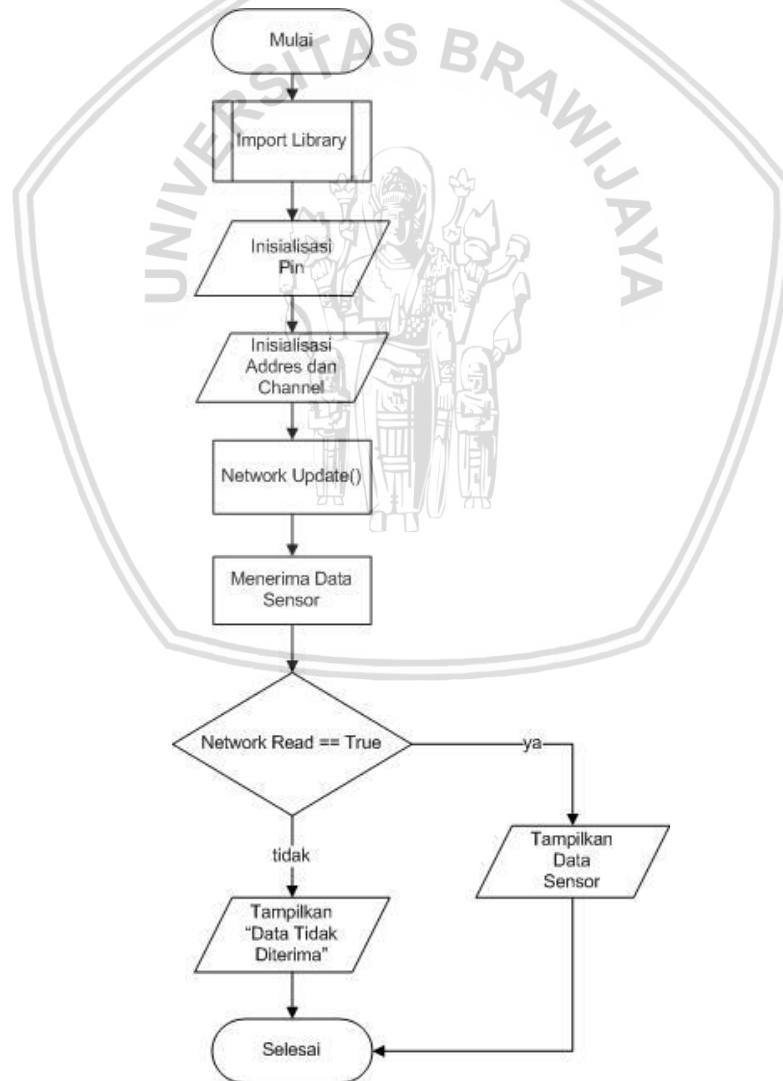


Gambar 5.7 Perancangan Perangkat Lunak *Node Transmitter*.

5.1.2.2 Perancangan Perangkat Lunak *Receiver*

Perancangan perangkat lunak pada *node receiver* menggunakan bahasa C yang diprogram menggunakan Arduino IDE. Pada perancangan ini diharapkan *node receiver* dapat menerima data sensor yang dikirim oleh *node transmitter*.

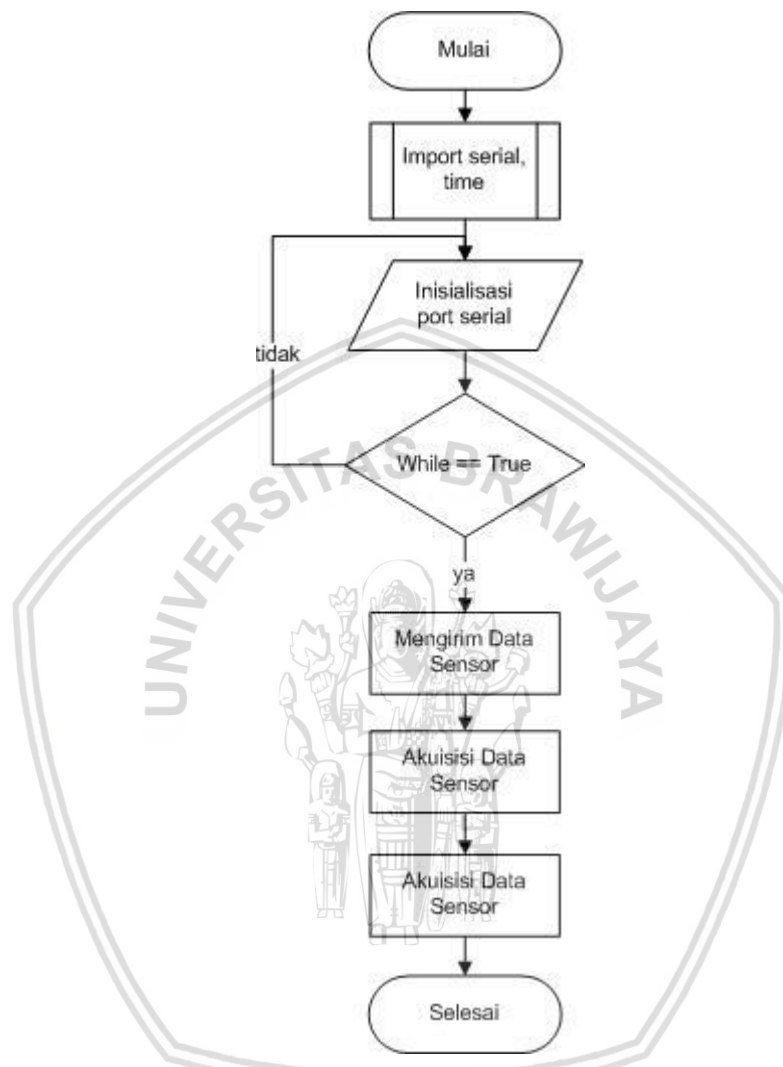
Sebelum *node receiver* menerima data dari *node transmitter* hal yang dilakukan juga sama dengan *node transmitter* yaitu import *library* yang dibutuhkan terlebih dahulu. Setelah itu melakukan inisialisasi pin yang digunakan oleh nRF24L01 dan diikuti dengan pemberian alamat *node* dan *channel*. Alamat yang digunakan oleh *node receiver* adalah 00 dan *channel* yang digunakan sama dengan *node transmitter* yaitu *channel* 90 agar nantinya data dari *node transmitter* dapat diterima oleh *node receiver*. *node receiver* akan mengecek apakah jaringan yang akan digunakan tersedia melalui perintah *network.update()*. Jika jaringan telah tersedia maka akan dilakukan proses listening, *node receiver* menunggu data yang dikirim dari *node transmitter*. Pada saat proses penerimaan data akan ada kondisi untuk pengecekan data. Jika data berhasil diterima maka data yang dikirim oleh *node transmitter* akan ditampilkan, baik itu count data, suhu dan kelembaban serta identitas *node* yang mengirimkan data sensor. Untuk perancangan perangkat lunak *receiver* dapat dilihat pada Gambar 5.8.



Gambar 5.8 Perancangan Perangkat Lunak *Node Receiver*.

5.1.3.2 Perancangan Perangkat Lunak Komunikasi Serial

Perancangan perangkat lunak komunikasi serial menggunakan bahasa pemrograman python, pemrograman bisa dilakukan menggunakan python GUI, CLI, maupun notepad++. Untuk perancangannya dapat dilihat pada Gambar 5.9.



Gambar 5.9 Diagram Alir Komunikasi Serial.

Dilihat dari diagram alir komunikasi serial untuk mendapatkan waktu dan pencatatan data sensor, terlebih dahulu melakukan import serial agar nantinya serial komunikasi dapat dikenali. Setelah melakukan import *library* selanjutnya melakukan inisialisasi port serial yang digunakan. Sebelum melakukan proses komunikasi serial terdapat kondisi untuk mengecek apakah komunikasi serial tersedia. Jika iya maka selanjutnya akan menjalankan proses untuk pembacaan komunikasi serial serta pengambilan waktu pc yang akan dicatat dalam file txt untuk memudahkan perhitungan maupun pencatatan data.

5.2 Impelementasi Sistem

Bab ini menjelaskan mengenai implementasi yang dilakukan dari perancangan pada point sebelumnya yang dimulai dari implementasi perangkat keras yang

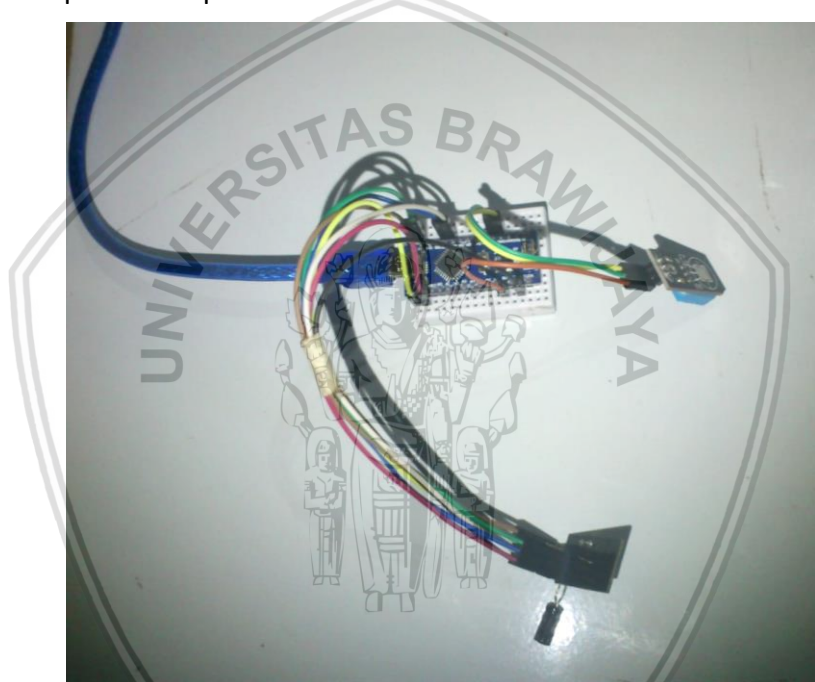
terdiri dari *node transmitter* dan *node receiver* serta implementasi perangkat lunak yang meliputi proses pengiriman, penerimaan data sensor serta komunikasi serial untuk pencatatan waktu dan data.

5.2.1 Implementasi Perangkat Keras

Implementasi perangkat keras terdapat 2 implementasi yang meliputi implementasi perangkat keras *transmitter* dan implementasi perangkat keras *receiver*.

5.2.1.1 Implementasi Perangkat Keras *Transmitter*

Implementasi *node transmitter* dilakukan sesuai dengan perancangan perangkat keras pada point sebelumnya. Implementasi perangkat keras *transmitter* dapat dilihat pada Gambar 5.10.

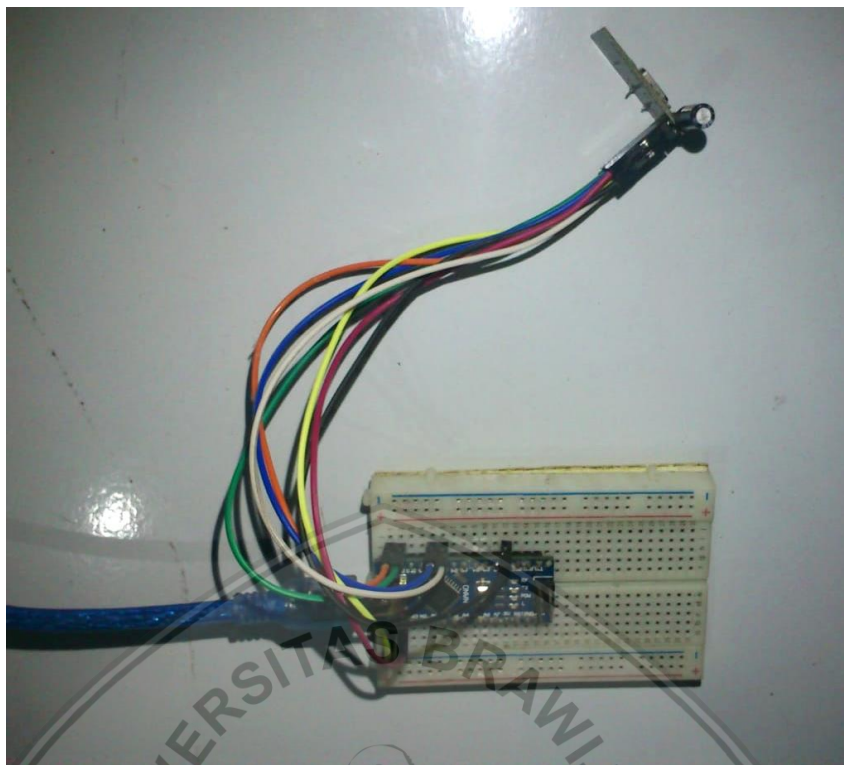


Gambar 5.10 Implementasi *Node Transmitter*

Dapat dilihat pada Gambar 5.10 perangkat keras *node transmitter* terdiri dari Arduino Nano sebagai Mikrokontroler, DHT11 sebagai sensor pendeteksi suhu dan kelembaban, dan nRF24L01 sebagai modul komunikasi wireless. Seluruh perangkat keras digabungkan menjadi satu agar dapat membentuk sebuah *node sensor* pada *node transmitter*.

5.2.1.2 Implementasi Perangkat Keras *Receiver*

Implementasi *node receiver* dilakukan sesuai dengan perancangan perangkat keras yang telah dilakukan point sebelumnya. Implementasi perangkat keras *receiver* dapat dilihat pada Gambar 5.11.



Gambar 5.11 Implementasi *Node Receiver*

Terlihat pada Gambar 5.11 perangkat keras *node receiver* terdiri dari Arduino Nano sebagai Mikrokontroler, dan nRF24L01 sebagai modul komunikasi wireless. Seluruh perangkat keras digabungkan menjadi satu agar dapat membentuk sebuah *node receiver*.

5.2.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada sistem ini menggunakan bahasa pemrograman C yang diprogram menggunakan Arduino IDE dan bahasa pemrograman python yang diprogram menggunakan notepad++. Dari implementasi perangkat lunak diharapkan *node transmitter* dapat mengakuisisi sensor suhu dan kelembaban serta dapat mengirimkan data sensor kepada *node receiver*. *Node receiver* dapat menerima data yang dikirim oleh *node transmitter*.

Agar dalam implementasi perangkat lunak tidak ada kendala, terlebih dahulu disarankan menyiapkan *library* serta *package* yang dibutuhkan, dan pada penelitian ini *library* serta *package* yang digunakan adalah RF24, RF24Network, DHT11 dan pySerial. pySerial digunakan untuk komunikasi serial, antara Python dengan Arduino. Yang nantinya digunakan untuk mendapatkan waktu pengiriman dan penerimaan data.

5.2.2.1 Implementasi Perangkat Lunak *Transmitter*

Agar *node transmitter* dapat mengirimkan data sensor ke *node receiver*, dibutuhkan implementasi perangkat lunak untuk menunjang implementasi dari perangkat keras yang dilakukan.

```

1 #include <RF24Network.h>
2 #include <RF24.h>
3 #include <DHT.h>
4 #include <SPI.h>

```

Agar perangkat keras dapat dikenali dan dapat bekerja sesuai dengan perintahnya, hal pertama yang dilakukan adalah import *library* yang dibutuhkan. Baris ke 1 import *library* untuk RF24Network, baris ke 2 import *library* untuk RF24, baris ke 3 import *library* untuk DHT11 dan baris ke 4 import *library* SPI untuk komunikasi SPI.

```

1 #define DHTPIN 2
2 #define DHTTYPE DHT11
3 DHT dht(DHTPIN, DHTTYPE);

```

Setelah dilakukan import *library* selanjutnya mendefinisikan pin dan type DHT yang digunakan. Baris ke 1 untuk mendefinisikan pin DHT11 yang akan digunakan di Mikrokontroler, baris ke 2 untuk mendefinisikan type DHT sensor dan baris ke 3 untuk memberikan informasi pin dan type yang digunakan.

```

1 RF24 radio(7,8);
2 RF24Network network(radio);
3 const uint16_t this_node = 01;
4 const unsigned long interval = 2000;
5 unsigned long last_sent;
6 long packet_sent=1;

```

Selanjutnya inisialisasi pin nRF24L01 yang digunakan di Mikrokontroler. Baris ke 1 untuk inisialisasi pin nRF24L01, baris ke 2 untuk mendeklarasikan RF24Network menggunakan komunikasi radio, baris ke3 untuk memberikan alamat *node*, baris ke 4 digunakan untuk inisialisasi interval pengiriman data, baris ke 5 untuk inisialisasi variable sent yang nantinya digunakan untuk kondisi interval pengiriman data, dan baris ke 6 untuk counting paket.

```

1 struct payload_t {
2     long counter;
3     float temperature;
4     float humidity;};

```

Selanjutnya pembuatan struc payloadnya untuk membungkus paket data yang akan dikirim nantinya sebagai object. Baris ke 1 untuk pembuatan struc objectnya, baris 2-4 untuk deklarasi variable data yang akan dikirimkan ke *node receiver*.

```

1 void setup(void) {
2   Serial.begin(9600);
3   Serial.println("RF24Network/Transmitter/");
4   SPI.begin();
5   radio.begin();
6   network.begin(/*channel*/ 90, /*node address*/ this_node);
7   dht.begin();
8 }

```

Setelah import *library*, inialisasi pin dan deklarasi *variable*. Selanjutnya ke inialisasi instruksi. Intruksi void setup hanya dijalankan sekali saat arduino mulai menyala. Baris ke 1 merupakan class dari inialisasi intruksi. Baris ke 2 untuk set baud rate yang digunakan untuk komunikasi. Baris ke 3 untuk print out identitas modul. Baris ke 4 untuk menjalankan komunikasi SPI. Baris ke 5 untuk menjalankan radio komunikasi. Baris ke 6 untuk set *channel* dan pemberian alamat *node* dan baris ke 7 untuk menjalankan sensor DHT11.

```

1 void loop(void) {
2   network.update();
3   unsigned long now = millis();
4   if ( now - last_sent >= interval ) {
5     last_sent = now;
6     float t = dht.readTemperature();
7     float h = dht.readHumidity();
8     payload_t payload = {packet_sent++, t, h};
9     RF24NetworkHeader header(/*to node*/ 00);
10    bool ok = network.write(header, &payload, sizeof(payload));
11    if (ok) {
12      Serial.print("Packet : ");
13      Serial.print(payload.counter);
14      Serial.print(" Temperature : ");
15      Serial.print(payload.temperature);
16      Serial.print(" *C Humidity : ");
17      Serial.print(payload.humidity);
18      Serial.print(" %");
19      Serial.println();
20    }
21  }
22 }

```

Sekarang masuk ke program inti, yang nantinya method ini akan dijalankan berulang-ulang. Baris ke 1 merupakan kelas methodnya. Baris ke 2 untuk pengecekan jaringan komunikasi apakah tersedia atau tidak. Baris ke 3 untuk deklarasi waktu dengan satuan milisecond yang nantinya akan digunakan untuk interval pengiriman paket data. Baris ke 4 merupakan kondisi untuk mengatur

pengiriman paket data berdasarkan interval waktu yang telah ditentukan. Baris ke 5 untuk mendapatkan waktu terbaru setelah data dikirim. Baris 6-7 untuk mendapatkan data sensor suhu dan kelembaban. Baris ke 8 untuk pembungkusan paket yang akan dikirim sesuai dengan format object. Baris ke 9 gunakan untuk header paket, paket yang dikirim akan diarahkan sesuai dengan alamat node tujuan. Baris 10-11 merupakan boolehan true false yang digunakan untuk mengecek paket apakah terkirim atau tidak ke node receiver. Baris 12-19 untuk mencetak print out jika pengecekan pengiriman paket terpenuhi dan 20-21 merupakan kondisi jika paket data tidak terkirim ke node receiver.

5.2.2.2 Implementasi Perangkat Lunak *Receiver*

Tidak jauh berbeda dengan *node transmitter*, agar *node receiver* mampu menerima data sesuai dengan fungsinya, dibutuhkan implementasi perangkat lunak untuk menunjang implementasi dari perangkat keras yang dilakukan.

```
1 #include <RF24Network.h>
2 #include <RF24.h>
3 #include <SPI.h>
```

Langkah pertama yang harus dilakukan untuk *node receiver* tidak jauh berbeda dengan *node transmitter*. Terlebih dahulu dilakukan import *library* yang dibutuhkan. Baris ke 1 import *library* untuk RF24Network, baris ke 2 import *library* untuk RF24, dan baris ke 3 untuk import *library* SPI untuk komunikasi SPI.

```
1 RF24 radio(7,8);
2 RF24Network network(radio);
3 const uint16_t this_node = 00;
```

Selanjutnya inisialisasi pin nRF24L01 yang digunakan di Mikrokontroler. Baris ke 1 untuk inisialisasi pin nRF24L01, baris ke 2 untuk mendeklarasikan RF24Network menggunakan komunikasi radio, dan baris ke 3 untuk memberikan alamat *node*.

```
1 struct payload_t {
2     long counter;
3     float temperature;
4     float humidity;
5 };
```

Selanjutnya pembuatan struct payloadnya untuk membungkus paket data yang akan diterima dari *node transmitter*. Baris ke 1 untuk pembuatan struct objectnya, baris 2-4 untuk deklarasi *variable* data yang akan digunakan untuk pengambilan paket data yang dikirim oleh *node transmitter*.


```

1 void setup(void) {
2     Serial.begin(9600);
3     Serial.println("RF24Network/Receiver/");
4     SPI.begin();
5     radio.begin();
6     network.begin(/*channel*/ 90, /*node address*/ this_node);
7 }

```

Untuk method instruksi tidak jauh berbeda dengan *node transmitter*. Baris ke 1 merupakan class dari inisialisasi intruksi. Baris ke 2 untuk set baud rate yang digunakan untuk komunikasi. Baris ke 3 untuk print out identitas modul. Baris ke 4 untuk menjalankan komunikasi SPI. Baris ke 5 untuk menjalankan radio komunikasi dan baris ke 6 untuk set *channel* dan pemberian alamat *node*.

```

1 void loop(void) {
2     network.update();
3     while (network.available()) {
4         RF24NetworkHeader header;
5         payload_t payload;
6         bool ok = network.read(header, &payload, sizeof(payload));
7         if (ok) {
8             Serial.print("Packet : ");
9             Serial.print(payload.counter);
10            Serial.print(" Temperature : ");
11            Serial.print(payload.temperature);
12            Serial.print(" *C Humidity : ");
13            Serial.print(payload.humidity);
14            Serial.print(" % Dari Node : ");
15            Serial.println(header.from_node);
16        }
17        else {
18            Serial.println("Data Sensor not receive");
19        }
20    }
21 }

```

Program inti node receiver tidak jauh berbeda dengan node transmitter, perbedaan terletak di baris ke 3 dan ke 6 saja. Baris ke 3 digunakan untuk mengecek apakah ada paket yang dikirimkan untuk node receiver. Jika ada maka akan menjalankan kondisi untuk pembacaan paket data yang terdapat pada baris ke 6.

5.2.2.3 Implementasi Perangkat Lunak Komunikasi Serial

Implementasi perangkat lunak komunikasi serial digunakan untuk membaca aktifitas yang terjadi pada serial komunikasi. komunikasi serial digunakan untuk mencatat waktu pengiriman dan penerimaan data yang nantinya digunakan untuk pengujian *delay*.

```
1 import serial,time
```

Untuk bisa menggunakan komunikasi serial serta mendapatkan waktu di pc/laptop hal pertama yang harus dilakukan adalah import *library* serial dan time.

```
1 port = serial.Serial('COM5', 9600)
2 write_to_file_path = "receiver.txt"
3 output_file = open(write_to_file_path, "a+")
```

Selanjutnya deklarasikan port serta beud rate yang digunakan. Baris 2-3 digunakan untuk menyimpan output kedalam text file.

```
1 while True:
2     line = port.readline()
3     b = str(time.time())
4     print b+'\t'+line
5     output_file.write((b)+'\t'+line)
6     output_file.flush()
```

Selanjutnya dilakukan pengecekan apakah port tersedia atau tidak. Jika tersedia maka akan menjalankan proses dari baris 2-6, yang digunakan untuk membaca aktifitas yang terjadi pada serial komunikasi, mendapatkan waktu pc/laptop dan mencatatnya kedalam file text yang sudah disediakan.

BAB 6 PENGUJIAN DAN ANALISIS

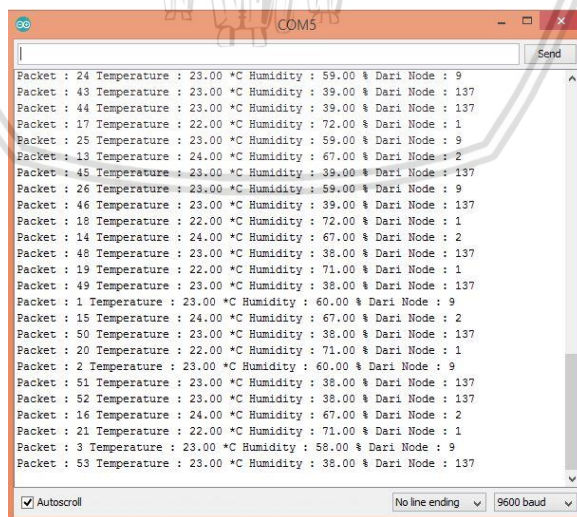
Bab ini membahas mengenai tahapan-tahapan pengujian dari implementasi sistem yang telah dilakukan yang meliputi skenario pengujian serta untuk mengetahui kinerja dari sistem yang diimplementasikan. Pengujian dan analisis bertujuan untuk mendapatkan kesimpulan dari implementasi kinerja sistem.

6.1 Pengujian Perangkat Keras

Pengujian perangkat keras bertujuan untuk mengetahui hasil dari implementasi yang dilakukan pada tahap sebelumnya. Dimana pengujian perangkat keras terdiri dari pengujian secara menyeluruh, baik dari implementasi *node transmitter* maupun implementasi *node receiver*. Untuk pengujian perangkat keras dapat dilihat pada Gambar 6.1 dan outputnya dapat dilihat pada Gambar 6.2.



Gambar 6.1 Pengujian Perangkat Keras



Gambar 6.2 Output dari Pengujian Perangkat Keras

Gambar 6.2 terlihat *node receiver* dapat menerima data yang dikirim oleh *node transmitter* yang menunjukkan bahwa implementasi sistem sudah berjalan dengan baik sesuai fungsinya.

6.2 Skenario Pengujian

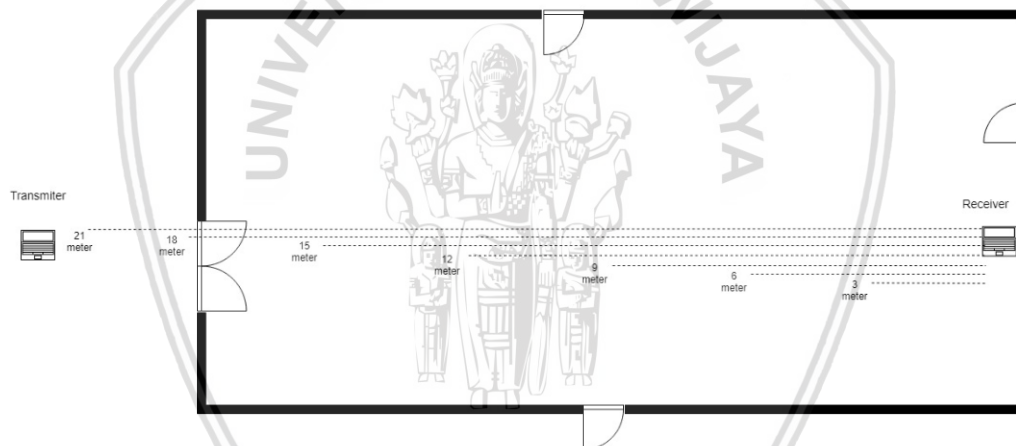
Skenario pengujian yang akan dilakukan pada penelitian ini dibagi menjadi 2 yaitu pengujian mekanisme *single-hop* dan pengujian mekanisme *multi-hop*. Setiap pengujian dilakukan pada 2 kondisi yang berbeda, kondisi pengiriman data tanpa penghalang dan kondisi pengiriman data dengan penghalang.

6.2.1 Pengujian Komunikasi 2 Node (*Single-hop*)

Pengujian *single-hop* terbagi menjadi 2 pengujian yaitu pengujian *single-hop* tanpa penghalang dan pengujian *single-hop* dengan penghalang. Pengujian komunikasi ini bertujuan untuk mengetahui jarak terjauh dari komunikasi antar *node*.

6.2.1.1 Pengujian Tanpa Penghalang

Pada percobaan pengujian 2 *node* tanpa penghalang, *node transmitter* akan mengirimkan 10 data kepada *node receiver* dengan rentang jarak kelipatan 3 meter dari letak *node receiver*. Denah pengujian untuk komunikasi 2 *node* dapat dilihat pada Gambar 6.3.



Gambar 6.3 Denah Pengujian 2 Node Tanpa Penghalang

1. Packet loss

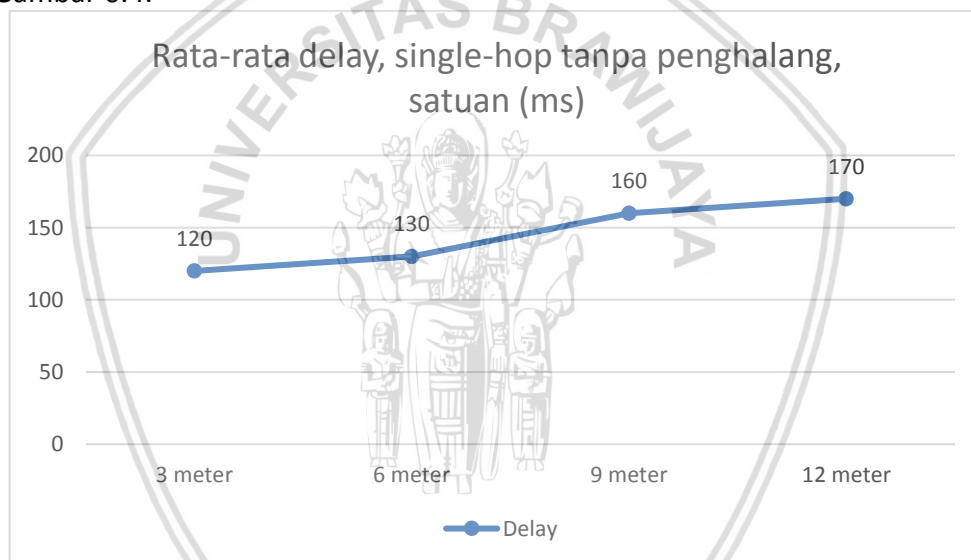
Berdasarkan hasil pengujian pengiriman data sensor dengan rentang jarak kelipatan 3 meter pada percobaan 2 *node* tanpa penghalang, terlihat pada jarak 15 meter sudah terdapat beberapa paket data yang tidak sampai ke *node receiver*. Untuk presentase *packet loss* pengiriman data dapat dilihat pada Tabel 6.1.

Tabel 6.1 *Packet loss* Mekanisme *Single-hop* Tanpa Penghalang

Jarak	Data Dikirm	Data Diterima	Packet Loss
3 meter	10	10	0%
6 meter	10	10	0%
9 meter	10	10	0%
12 meter	10	10	0%
15 meter	10	7	30%
18 meter	10	4	60%
21 meter	10	0	100%

2. Delay

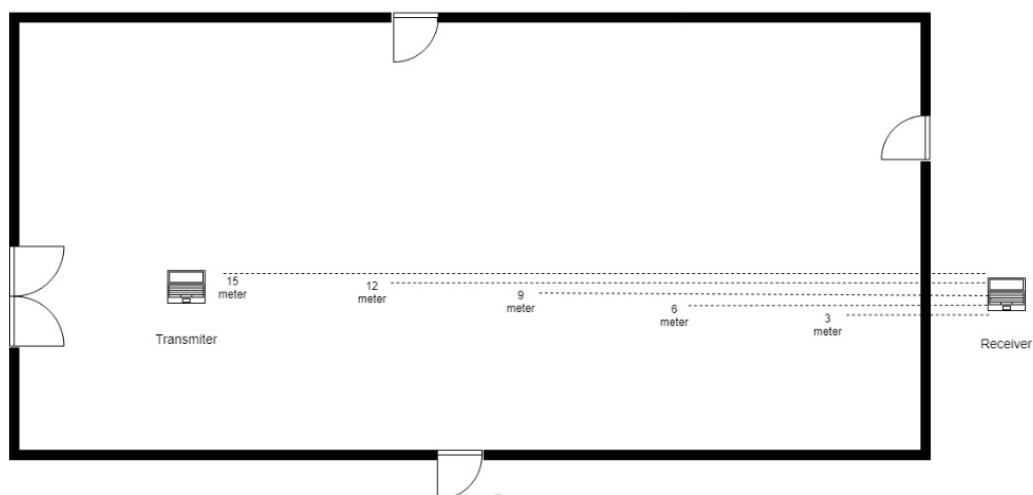
Dari hasil percobaan pengujian pengiriman data antar *node* tanpa penghalang terlihat pada jarak 15 meter paket data yang dikirim sudah banyak yang tidak sampai ke *node receiver*. Maka diambil *delay* pengiriman paket data untuk pengujian dari jarak 3 meter sampai 12 meter yang dapat dilihat pada Gambar 6.4.



Gambar 6.4 Rata-rata *Delay* Mekanisme *Single-hop* Tanpa Penghalang

6.2.1.2 Pengujian Dengan Penghalang

Percobaan pengujian 2 *node* dengan penghalang hampir sama dengan percobaan 2 *node* tanpa penghalang, *node transmitter* akan mengirimkan 10 data kepada *node receiver* dengan rentang jarak kelipatan 3 meter dari letak *node receiver*. Denah pengujian untuk komunikasi 2 *node* dengan penghalang dapat dilihat pada Gambar 6.5.



Gambar 6.5 Denah Pengujian 2 *Node* dengan Penghalang

1. *Packet loss*

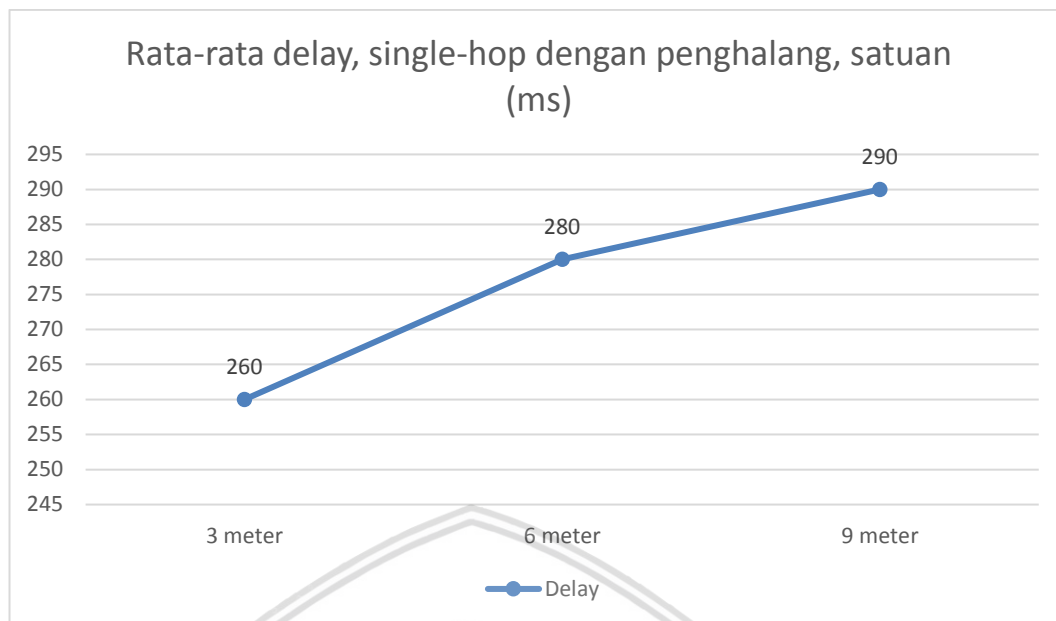
Berdasarkan hasil pengujian pengiriman data sensor antar *node* dengan rentang jarak kelipatan 3 meter dengan penghalang, terlihat pada jarak 12 meter sudah terdapat beberapa paket data yang tidak sampai ke *node receiver*. Untuk presentase *packet loss* pengiriman data dapat dilihat pada Tabel 6.2.

Tabel 6.2 *Packet loss* Mekanisme *Single-hop* dengan Penghalang

Jarak	Data Dikirim	Data Diterima	Packet Loss
3 meter	10	10	0%
6 meter	10	10	0%
9 meter	10	10	0%
12 meter	10	6	40%
15 meter	10	3	70%
18 meter	10	0	100%

2. *Delay*

Dari hasil percobaan pengujian pengiriman data antar *node* terlihat pada jarak 12 meter paket data sudah banyak yang tidak sampai ke *node receiver*. Maka diambil *delay* pengiriman paket data untuk pengujian dari jarak 3 meter sampai 9 meter. *Delay* pengiriman paket dari jarak 3 meter sampai 9 meter dapat dilihat pada Gambar 6.6.



Gambar 6.6 Rata-rata *Delay* Mekanisme *Single-hop* dengan Penghalang

6.2.2 Pengujian Mekanisme *Multi-hop*

Pengujian mekanisme *multi-hop* terbagi menjadi 2 pengujian yaitu pengujian *multi-hop* tanpa penghalang dan pengujian *multi-hop* dengan penghalang. Pengujian mekanisme *multi-hop* merupakan pengujian lanjutan dari mekanisme *single-hop*, pada mekanisme *single-hop* dengan percobaan pengiriman tanpa penghalang pada jarak 15 meter paket data yang dikirim ke *node receiver* data sudah banyak yang hilang sedangkan pada percobaan dengan penghalang pada jarak 12 meter data sudah banyak yang hilang. Maka untuk percobaan pengiriman data menggunakan mekanisme *multi-hop* dengan percobaan pengiriman tanpa penghalang akan dilakukan dengan jarak 12 meter antar *node* dan percobaan dengan penghalang akan dilakukan dengan jarak 9 meter antar *node*.

6.2.2.1 Pengujian Mekanisme 1 Hop (3 Node)

Pengujian mekanisme 1 hop membutuhkan 3 *node*, 1 *node* sebagai *node relay* dan *node* lainnya sebagai *transmitter* dan *receiver*. *Node relay* juga berperan untuk mengirimkan data sensor kepada *node receiver*. Pengujian mekanisme 1 hop dibagi menjadi 2 pengujian, yaitu pengujian tanpa penghalang dan pengujian dengan penghalang.

1. Pengujian Tanpa Penghalang

Pada percobaan pengujian 1 hop tanpa penghalang, *node transmitter* akan mengirimkan 10 data kepada *node receiver* dengan rentang jarak 12 meter antar *node*. Denah pengujian mekanisme 1 hop tanpa penghalang dapat dilihat pada Gambar 6.7.



Gambar 6.7 Denah Pengujian Mekanisme 1 Hop Tanpa Penghalang

a. *Packet loss*

Berdasarkan hasil pengujian pengiriman data sensor dengan jarak 12 meter pada percobaan 1 hop tanpa penghalang terlihat data sensor dari *node transmitter* dapat terkirim dengan baik. jarak total dari *node transmitter* dengan *node receiver* adalah 24 meter. Yang sebelumnya pada mekanisme *single-hop* tanpa penghalang pada jarak 15 meter data yang dikirim sudah banyak yang tidak sampai ke *node receiver*. Output mekanisme 1 hop tanpa penghalang dapat dilihat pada Gambar 6.8 dan presentase *packet loss* dapat dilihat pada Tabel 6.3.

```

1532441162.23 Packet : 1 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441164.23 Packet : 2 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441164.55 Packet : 4 Temperature : 23.00 *C Humidity : 72.00 % Dari Node : 1
1532441166.23 Packet : 3 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441167.55 Packet : 5 Temperature : 23.00 *C Humidity : 72.00 % Dari Node : 1
1532441168.23 Packet : 4 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441170.23 Packet : 5 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441170.56 Packet : 6 Temperature : 23.00 *C Humidity : 72.00 % Dari Node : 1
1532441172.23 Packet : 6 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441173.56 Packet : 7 Temperature : 23.00 *C Humidity : 72.00 % Dari Node : 1
1532441174.23 Packet : 7 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441176.23 Packet : 8 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441178.23 Packet : 9 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
1532441180.23 Packet : 10 Temperature : 23.00 *C Humidity : 81.00 % Dari Node : 9
    
```

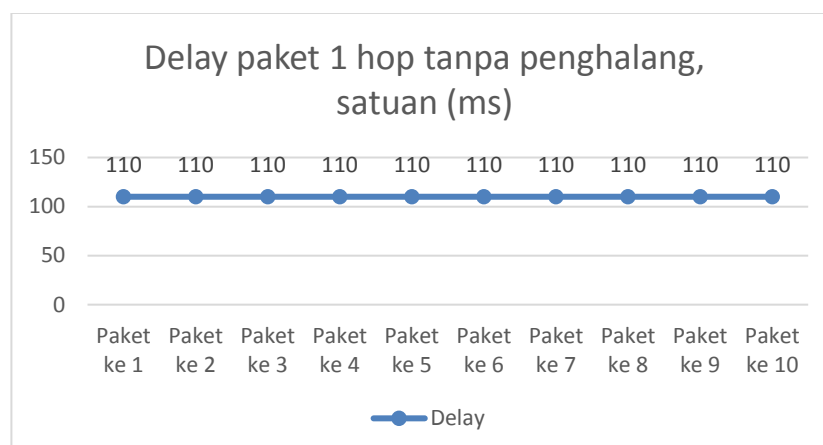
Gambar 6.8 Mekanisme 1 Hop Tanpa Penghalang

Tabel 6.3 *Paket Loss* 1 Hop Tanpa Penghalang

Jarak	Data Dikirim	Data Diterima	Packet Loss
24 meter	10	10	0%

b. *Delay*

Dari hasil percobaan pengujian pengiriman data menggunakan 1 hop pada jarak 12 meter antar *node* dengan 10 data yang dikirim. Didapatkan *delay* pengiriman data dari *node transmitter* menuju *node receiver* yang dapat dilihat pada Gambar 6.9.

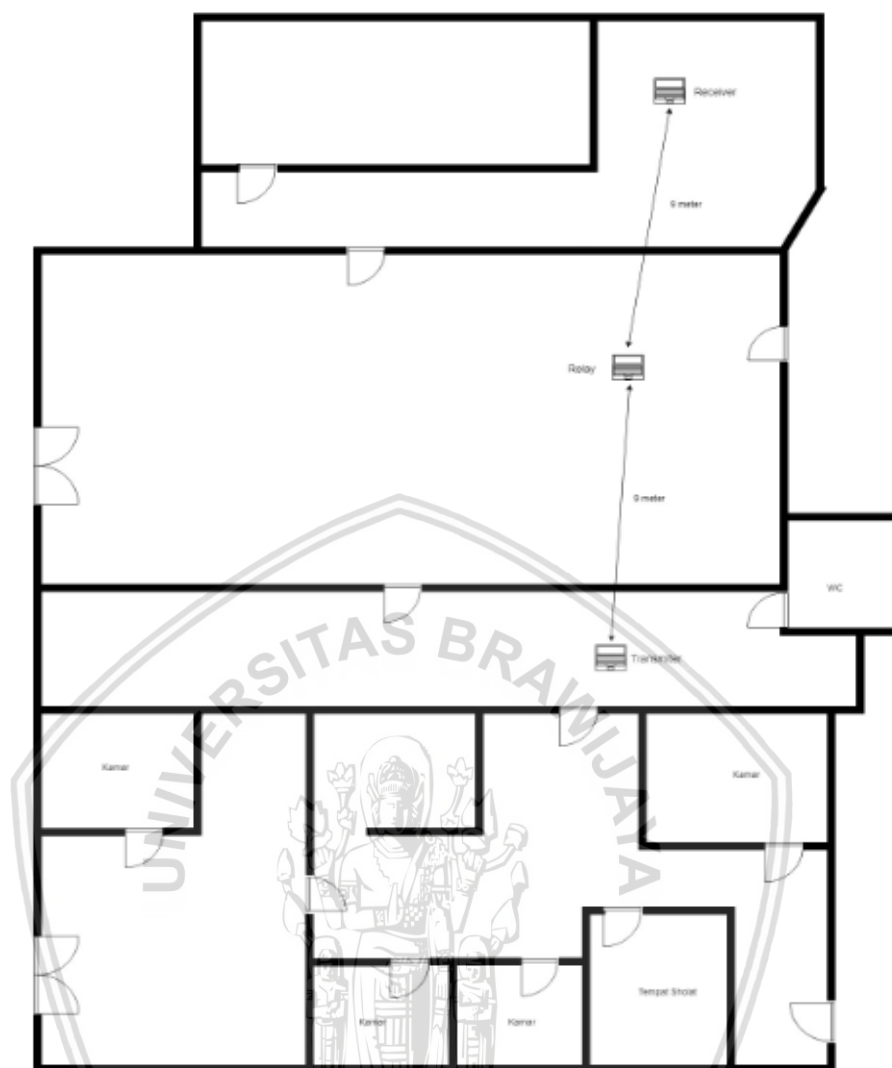


Gambar 6.9 Delay Paket 1 Hop Tanpa Penghalang

Terlihat pada Gambar 6.9 *delay* pengiriman data pada mekanisme 1 hop cenderung memiliki *delay* pengiriman data yang stabil, dari 10 data yang dikirim memiliki *delay* paket yang sama. Dari Gambar 6.9 didapatkan rata-rata *delay* pengiriman paket sebesar 110 ms.

2. Pengujian Dengan Penghalang

Percobaan pengujian 1 hop dengan penghalang hampir sama dengan pengujian 1 hop tanpa penghalang, yang membedakan hanya jarak antar nodenya. Pada pengujian 1 hop dengan penghalang jarak antar *node* adalah 9 meter. Denah pengujian untuk mekanisme 1 hop dengan penghalang dapat dilihat pada Gambar 6.10.



Gambar 6.10 Denah Pengujian 1 Hop dengan Penghalang

a. *Packet loss*

Berdasarkan hasil pengujian pengiriman data sensor dengan jarak 9 meter antar *node* pada percobaan 1 hop dengan penghalang terlihat data sensor dari *node transmitter* dapat terkirim dengan baik ke *node receiver*. jarak total dari *node transmitter* dengan *node receiver* adalah 18 meter, yang sebelumnya pada mekanisme *single-hop* dengan penghalang pada jarak 12 meter data yang dikirim ke *node receiver* sudah banyak yang hilang. Output mekanisme 1 hop dengan penghalang dapat dilihat pada Gambar 6.11 dan presentase *packet loss* dapat dilihat pada Tabel 6.4.

```
c:\Python27>receive.py
1532317652.6  ARF24Network/Receiver/
1532317655.14  Packet : 2 Temperature : 29.00 *C Humidity : 52.00 % Dari Node : 1
1532317655.82  Packet : 1 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317657.82  Packet : 2 Temperature : 30.00 *C Humidity : 47.00 % Dari Node : 9
1532317658.15  Packet : 3 Temperature : 29.00 *C Humidity : 52.00 % Dari Node : 1
1532317659.83  Packet : 3 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317661.15  Packet : 4 Temperature : 29.00 *C Humidity : 52.00 % Dari Node : 1
1532317661.83  Packet : 4 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317663.83  Packet : 5 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317665.83  Packet : 6 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317667.85  Packet : 7 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317668.77  Packet : 1 Temperature : 29.00 *C Humidity : 52.00 % Dari Node : 1
1532317669.83  Packet : 8 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317671.79  Packet : 2 Temperature : 29.00 *C Humidity : 52.00 % Dari Node : 1
1532317671.85  Packet : 9 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532317673.83  Packet : 10 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
```

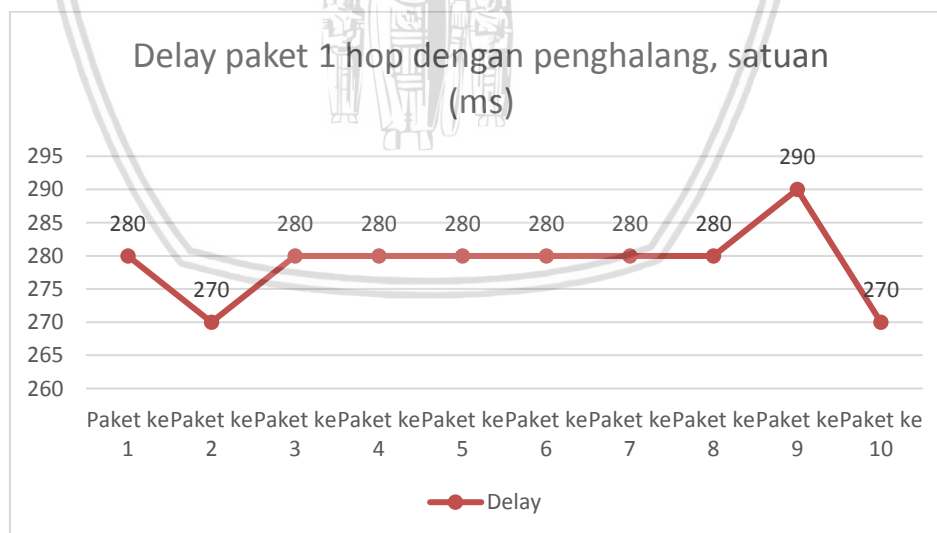
Gambar 6.11 Mekanisme 1 Hop dengan Penghalang

Tabel 6.4 Packet loss 1 Hop dengan Penghalang

Jarak	Data Dikirim	Data Diterima	Packet Loss
18 meter	10	10	0%

b. Delay

Dari hasil percobaan pengujian pengiriman data menggunakan 1 hop pada jarak 9 meter antar node dengan 10 data yang dikirim. Didapatkan delay pengiriman data dari node transmitter menuju node receiver dapat dilihat pada Gambar 6.12.



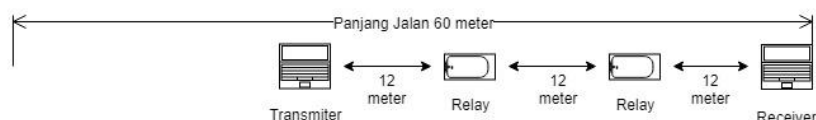
Gambar 6.12 Delay Paket 1 Hop dengan Penghalang

Terlihat pada Gambar 6.12 delay pengiriman data pada mekanisme 1 hop terlihat tidak stabil, terlihat pada paket ke 2, ke 9 dan ke 10 mempunyai perbedaan delay pengiriman data yang cukup jauh. Dari Gambar 6.12 didapatkan rata-rata delay pengiriman data sebesar 280 ms.

6.2.2.2 Pengujian Mekanisme 2 Hop (4 Node)

1. Pengujian Tanpa Penghalang

Pada percobaan pengujian 2 hop tanpa penghalang, *node transmitter* akan mengirimkan 10 data kepada *node receiver* dengan rentang jarak 12 meter dari letak *node receiver*. Denah pengujian mekanisme 2 hop tanpa penghalang dapat dilihat pada Gambar 6.13.



Gambar 6.13 Denah Pengujian 2 Hop Tanpa Penghalang

a. Packet loss

Berdasarkan hasil pengujian pengiriman data sensor dengan jarak 12 meter antar *node* terlihat data sensor dari *node transmitter* dapat terkirim dengan baik ke *node receiver*. jarak total dari *node transmitter* dengan *node receiver* adalah 36 meter. Output mekanisme 2 hop tanpa penghalang dapat dilihat pada Gambar 6.5 dan presentase *packet loss* dapat dilihat pada Tabel 6.5.

```

1532441819.03 Packet : 1 Temperature : 25.00 *C Humidity : 48.00 % Dari Node : 73
1532441820.31 Packet : 12 Temperature : 24.00 *C Humidity : 70.00 % Dari Node : 9
1532441821.03 Packet : 2 Temperature : 25.00 *C Humidity : 48.00 % Dari Node : 73
1532441822.22 Packet : 13 Temperature : 23.00 *C Humidity : 56.00 % Dari Node : 1
1532441823.04 Packet : 3 Temperature : 25.00 *C Humidity : 47.00 % Dari Node : 73
1532441823.31 Packet : 13 Temperature : 24.00 *C Humidity : 70.00 % Dari Node : 9
1532441825.04 Packet : 4 Temperature : 25.00 *C Humidity : 47.00 % Dari Node : 73
1532441826.22 Packet : 14 Temperature : 23.00 *C Humidity : 56.00 % Dari Node : 1
1532441826.31 Packet : 14 Temperature : 24.00 *C Humidity : 70.00 % Dari Node : 9
1532441827.04 Packet : 5 Temperature : 25.00 *C Humidity : 46.00 % Dari Node : 73
1532441829.32 Packet : 15 Temperature : 24.00 *C Humidity : 70.00 % Dari Node : 9
1532441829.39 Packet : 6 Temperature : 25.00 *C Humidity : 46.00 % Dari Node : 73
1532441830.22 Packet : 15 Temperature : 23.00 *C Humidity : 66.00 % Dari Node : 1
1532441831.05 Packet : 7 Temperature : 25.00 *C Humidity : 46.00 % Dari Node : 73
1532441832.32 Packet : 16 Temperature : 24.00 *C Humidity : 70.00 % Dari Node : 9
1532441833.05 Packet : 8 Temperature : 25.00 *C Humidity : 45.00 % Dari Node : 73
1532441834.21 Packet : 16 Temperature : 23.00 *C Humidity : 57.00 % Dari Node : 1
1532441835.33 Packet : 17 Temperature : 24.00 *C Humidity : 70.00 % Dari Node : 9
1532441835.39 Packet : 9 Temperature : 25.00 *C Humidity : 45.00 % Dari Node : 73
1532441837.06 Packet : 10 Temperature : 25.00 *C Humidity : 45.00 % Dari Node : 73
    
```

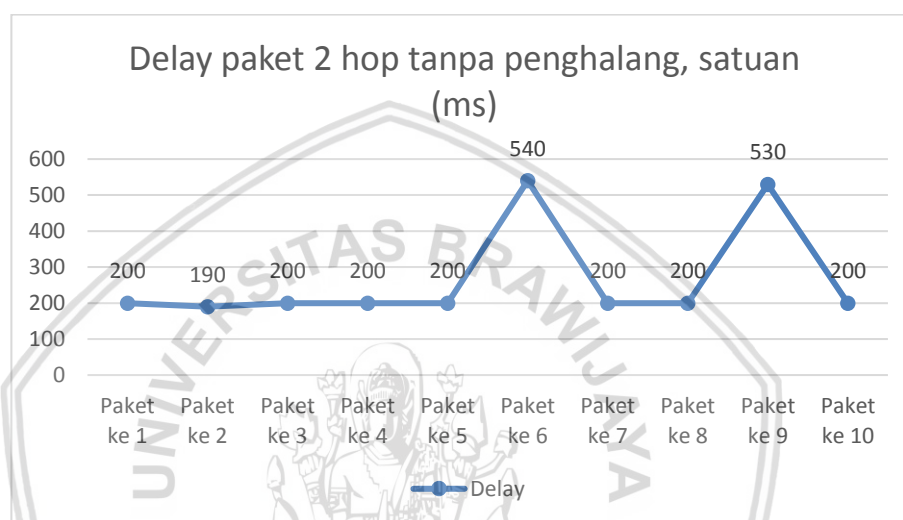
Gambar 6.5 Mekanisme 2 Hop Tanpa Penghalang

Tabel 6.5 *Packet loss 2 Hop Tanpa Penghalang*

Jarak	Data Dikirim	Data Diterima	Packet Loss
36 meter	10	10	0%

b. *Delay*

Dari hasil percobaan pengujian pengiriman data menggunakan 2 hop pada jarak 12 meter antar *node* dengan 10 data yang dikirim, didapatkan *delay* pengiriman paket data dari *node transmitter* menuju *node receiver* yang dapat dilihat pada Gambar 6.14.

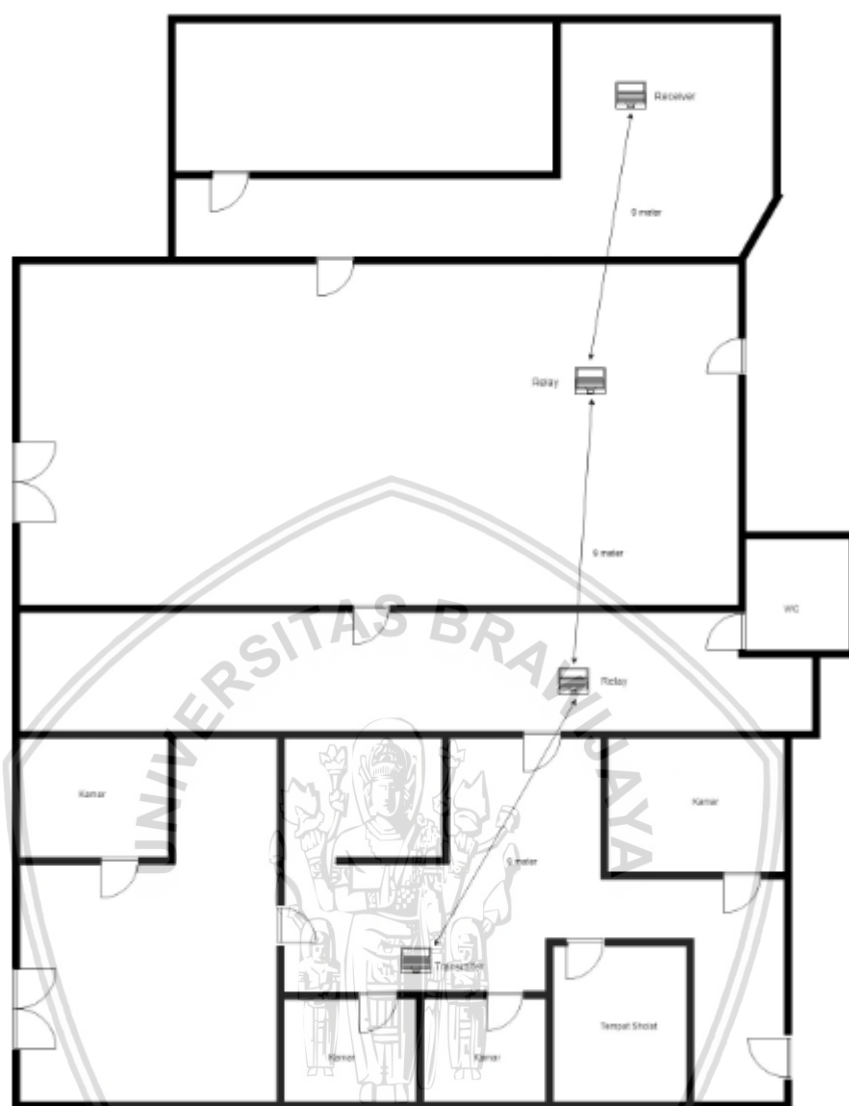


Gambar 6.14 *Delay Paket 2 Hop Tanpa Penghalang*

Terlihat pada Gambar 6.14 *delay* pengiriman data pada 2 hop terlihat tidak stabil, terlihat pada paket ke 6, dan paket ke 9 mempunyai perbedaan *delay* pengiriman yang jauh. Dari Gambar 6.14 didapatkan rata-rata *delay* pengiriman paket data sebesar 270 ms.

2. Pengujian Dengan Penghalang

Percobaan pengujian 2 hop dengan penghalang hampir sama dengan pengujian 2 hop dengan penghalang, *node transmitter* akan mengirimkan 10 data kepada *node receiver* dengan rentang jarak antar *node* 9 meter, jarak total dari *node transmitter* dengan *node receiver* adalah 27 meter. Denah pengujian untuk mekanisme 2 hop dengan penghalang dapat dilihat pada Gambar 6.15.



Gambar 6.15 Denah Pengujian 2 Hop dengan Penghalang

a. *Packet loss*

Berdasarkan hasil pengujian pengiriman data sensor dengan jarak total dari *node transmitter* sampai *node receiver* adalah 27 meter. Terlihat data sensor dari *node transmitter* dapat terkirim dengan baik ke *node receiver*. Output mekanisme 2 hop dengan penghalang dapat dilihat pada Gambar 6.16 dan untuk presentase *packet loss* dapat dilihat pada Tabel 6.6.

```
c:\Python27>receive.py
1532318606.56 RF24Network/Receiver/
1532318609.17 Packet : 222 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318610.16 Packet : 2 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
1532318610.23 Packet : 1 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318612.17 Packet : 223 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318612.25 Packet : 2 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318614.16 Packet : 3 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
1532318614.23 Packet : 3 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318615.17 Packet : 224 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318616.16 Packet : 4 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318618.17 Packet : 4 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
1532318618.23 Packet : 225 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318618.3 Packet : 5 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318620.17 Packet : 6 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318621.17 Packet : 226 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318622.17 Packet : 5 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
1532318622.23 Packet : 7 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318624.19 Packet : 227 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318624.25 Packet : 8 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318626.17 Packet : 6 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
1532318626.23 Packet : 9 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318627.17 Packet : 228 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532318628.17 Packet : 10 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532318630.17 Packet : 7 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
```

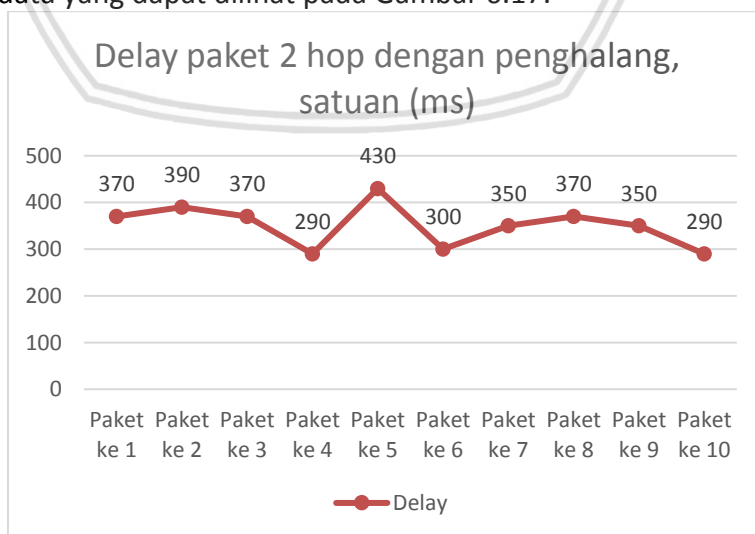
Gambar 6.16 Mekanisme 2 Hop dengan Penghalang

Tabel 6.6 Packet loss 2 Hop dengan Penghalang

Jarak	Data Dikirim	Data Diterima	Packet Loss
27 meter	10	10	0%

b. Delay

Dari hasil pengiriman data dari *node transmitter* menuju *node receiver* pada percobaan 2 hop dengan penghalang, didapatkan *delay* pengiriman paket data yang dapat dilihat pada Gambar 6.17.



Gambar 6.17 Delay Paket 2 Hop dengan Penghalang.

Terlihat pada Gambar 6.17 *delay* pengiriman data pada 2 *hop* dengan penghalang terlihat tidak stabil, terlihat pada paket ke 4, 5 dan 6 mempunyai perbedaan *delay* pengiriman yang cukup jauh. Dari Gambar 6.17 didapatkan rata-rata *delay* pengiriman paket sebesar 350 ms.

6.2.2.3 Pengujian Mekanisme 3 Hop (5 Node)

1. Pengujian Tanpa Penghalang

Pada percobaan pengujian 3 *hop* tanpa penghalang, *node transmitter* akan mengirimkan 10 data kepada *node receiver* dengan rentang jarak 12 meter dari letak *node receiver*. Denah pengujian mekanisme 2 *hop* tanpa penghalang dapat dilihat pada Gambar 6.18.



Gambar 6.18 Denah Pengujian 3 Hop Tanpa Penghalang

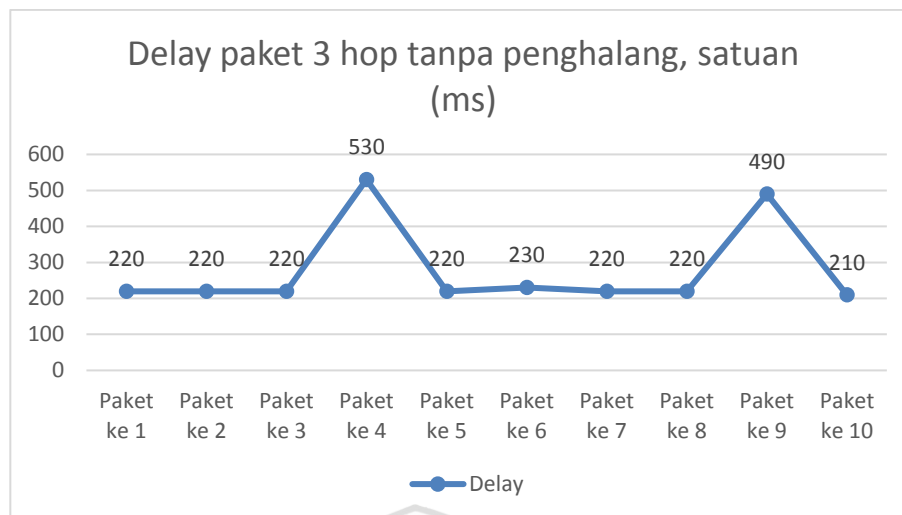
a. Packet loss

Berdasarkan hasil pengujian pengiriman data sensor dengan jarak 12 meter antar *node* terlihat data sensor dari *node transmitter* dapat terkirim dengan baik ke *node receiver*. jarak total dari *node transmitter* dengan *node receiver* adalah 48 meter. Output mekanisme 3 *hop* tanpa penghalang dapat dilihat pada Gambar 6.19 dan presentase *packet loss* dapat dilihat pada Tabel 6.7.

Gambar 6.19 Mekanisme 3 *Hop* Tanpa Penghalang.

Jarak	Data Dikirim	Data Diterima	Packet Loss
48 meter	10	10	0%

Dari hasil percobaan pengujian pengiriman data menggunakan 3 hop tanpa penghalang pada jarak 12 meter antar *node* dengan 10 data yang dikirim, didapatkan *delay* pengiriman paket data dari *node transmitter* menuju *node receiver* dapat dilihat pada Gambar 6.20.

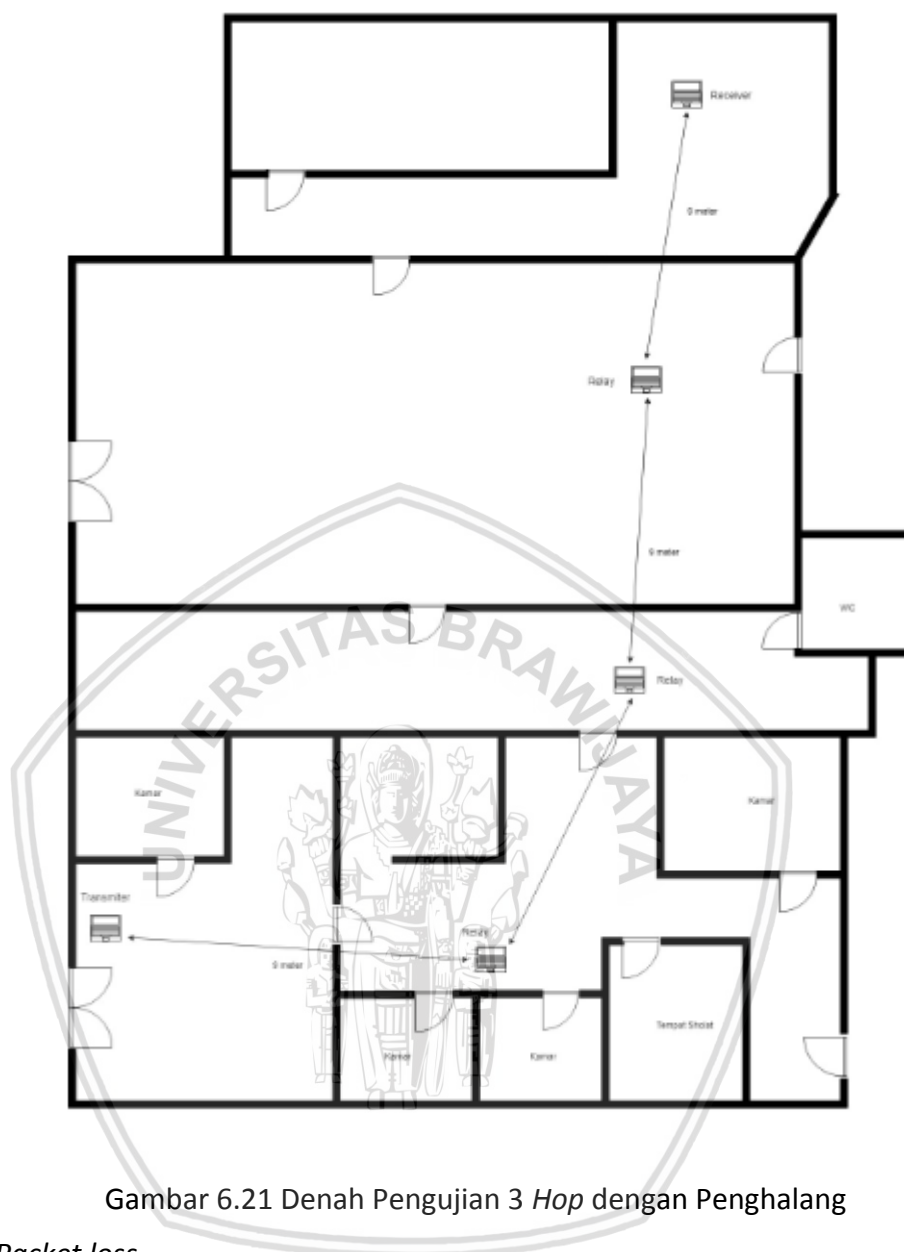


Gambar 6.20 Delay Paket 3 Hop Tanpa Penghalang

Terlihat pada Gambar 6.20 *delay* pengiriman data pada mekanisme 3 hop dengan penghalang terlihat tidak stabil, terlihat pada paket ke 4, dan paket ke 9 mempunyai perbedaan *delay* pengiriman yang jauh. Dari Gambar 6.20 didapatkan rata-rata *delay* pengiriman paket data sebesar 280 ms.

2. Pengujian Dengan Penghalang

Pengujian 3 hop dengan penghalang hampir sama dengan pengujian 3 hop tanpa penghalang, yang membedakan hanya jarak antar *node*. jarak total *node transmitter* dengan *node receiver* pada mekanisme 3 hop adalah 36 meter. Denah pengujian untuk mekanisme 3 hop dengan penghalang dapat dilihat pada Gambar 6.21.



Gambar 6.21 Denah Pengujian 3 Hop dengan Penghalang

a. *Packet loss*

Berdasarkan hasil pengujian pengiriman data sensor dengan jarak total *node transmitter* sampai *node receiver* sejauh 36 meter, data sensor dapat terkirim dengan baik. Yang sebelumnya pada mekanisme *single-hop* dengan penghalang pada jarak 12 meter sudah terdapat banyak data yang hilang saat pengiriman data. Output mekanisme 3 hop dengan penghalang dapat dilihat pada Gambar 6.22 dan presentase *packet loss* dapat dilihat pada Tabel 6.8.

```
c:\Python27>receive.py
1532319273.62 RF24Network/Receiver/
1532319273.88 Packet : 29 Temperature : 29.00 *C Humidity : 48.00 % Dari Node : 585
1532319275.78 Packet : 54 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532319276.7 Packet : 1 Temperature : 29.00 *C Humidity : 47.00 % Dari Node : 1
1532319277.23 Packet : 53 Temperature : 30.00 *C Humidity : 45.00 % Dari Node : 9
1532319278.2 Packet : 1 Temperature : 29.00 *C Humidity : 48.00 % Dari Node : 585
1532319278.76 Packet : 55 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532319280.2 Packet : 2 Temperature : 29.00 *C Humidity : 50.00 % Dari Node : 585
1532319281.23 Packet : 54 Temperature : 30.00 *C Humidity : 45.00 % Dari Node : 9
1532319281.71 Packet : 2 Temperature : 29.00 *C Humidity : 47.00 % Dari Node : 1
1532319281.78 Packet : 56 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532319282.21 Packet : 3 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319284.2 Packet : 4 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319285.23 Packet : 55 Temperature : 30.00 *C Humidity : 45.00 % Dari Node : 9
1532319286.21 Packet : 5 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319287.76 Packet : 58 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532319288.21 Packet : 6 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319289.18 Packet : 1 Temperature : 29.00 *C Humidity : 46.00 % Dari Node : 1
1532319289.24 Packet : 56 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 9
1532319290.21 Packet : 7 Temperature : 29.00 *C Humidity : 48.00 % Dari Node : 585
1532319290.78 Packet : 59 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532319292.21 Packet : 8 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319293.23 Packet : 57 Temperature : 30.00 *C Humidity : 45.00 % Dari Node : 9
1532319293.78 Packet : 60 Temperature : 29.00 *C Humidity : 13.00 % Dari Node : 73
1532319294.18 Packet : 2 Temperature : 30.00 *C Humidity : 46.00 % Dari Node : 1
1532319294.24 Packet : 9 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319296.22 Packet : 10 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
1532319298.22 Packet : 11 Temperature : 29.00 *C Humidity : 49.00 % Dari Node : 585
```

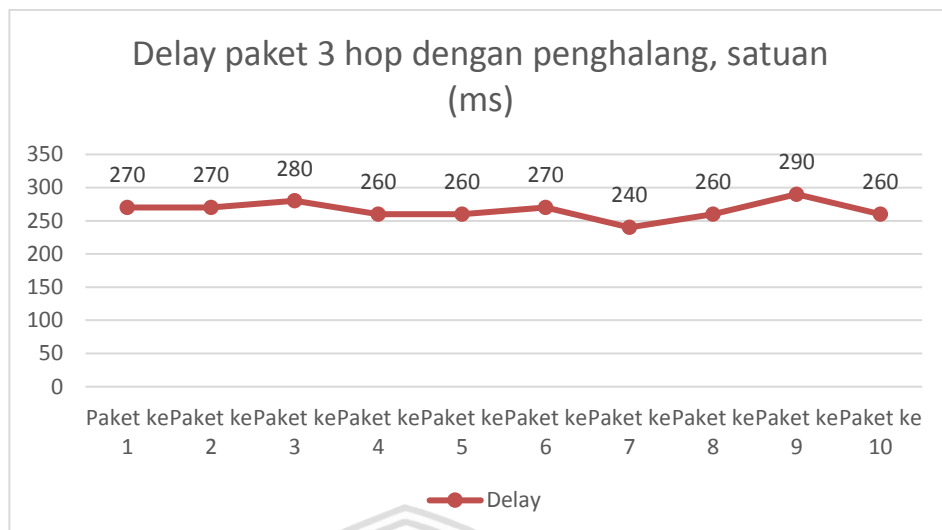
Gambar 6.22 Mekanisme 3 Hop dengan Penghalang.

Tabel 6.8 Packet loss 3 Hop dengan Penghalang

Jarak	Data Dikirim	Data Diterima	Packet Loss
36 meter	10	10	0%

b. Delay

Dari hasil percobaan pengujian pengiriman data menggunakan 3 hop pada jarak 9 meter antar *node* dengan 10 data yang dikirim didapatkan *delay* pengiriman paket dari *node transmitter* menuju *node receiver* dapat dilihat pada Gambar 6.23.



Gambar 6.23 Delay Paket 3 Hop dengan Penghalang

Terlihat pada Gambar 6.23 *delay* pengiriman data pada mekanisme 3 *hop* dengan penghalang terlihat stabil, range pengiriman *delay* dari paket 1 sampai paket 10 tidak terlalu jauh jaraknya. Dari Gambar 6.23 didapatkan rata-rata *delay* pengiriman paket data sebesar 260 ms.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan dari perancangan sistem, implementasi sistem, pengujian sistem dan analisis sistem dapat diambil kesimpulan antara lain:

1. Agar antar *node* dapat berkomunikasi menggunakan modul nRF24L01 dengan protokol RF24Network dibutuhkan library RF24.h, RF24Network.h dan SPI.h
2. Untuk mekanisme *multi-hop* agar *node transmitter* dapat mengirimkan data sensor ke *node receiver* dibutuhkan pengalamatan yang sesuai dengan *node* clusternya.
3. Mekanisme *single-hop* didapatkan tingkat akurasi pengiriman data dan total rata-rata *delay* sebagai berikut:
 - Tanpa penghalang jarak 3-12 meter akurasi pengiriman data mencapai 100%, 15 meter 70%, 18 meter 40%, 21 meter 0% dengan total rata-rata *delay* sebesar 145 ms,
 - Dengan penghalang jarak 3-9 meter akurasi pengiriman data mencapai 100%, 12 meter 60%, 15 meter 30%, 18 meter 0% dengan total rata-rata *delay* sebesar 276 ms.
4. Mekanisme *multi-hop* didapatkan tingkat akurasi pengiriman data dan total rata-rata *delay* sebagai berikut:
 - 1 *hop* tanpa penghalang jarak antar node 12 meter akurasi pengiriman data mencapai 100% dengan rata-rata *delay* sebesar 110 ms, jarak antar node 9 meter mencapai 100% dengan rata-rata *delay* sebesar 270 ms.
 - 2 *hop* tanpa penghalang jarak antar node 12 meter akurasi pengiriman data mencapai 100% dengan rata-rata *delay* sebesar 270 ms, jarak antar node 9 meter mencapai 100% dengan rata-rata *delay* sebesar 350 ms.
 - 3 *hop* tanpa penghalang jarak antar node 12 meter akurasi pengiriman data mencapai 100% dengan rata-rata *delay* sebesar 280 ms, jarak antar node 9 meter mencapai 100% dengan rata-rata *delay* sebesar 260 ms.
5. Dari segi pengukuran kinerja jaringan untuk pemantauan kondisi ruangan, tanpa adanya penghalang lebih unggul dibandingkan dengan adanya penghalang. Pengukuran kinerja berdasarkan *packet loss* dan *delay*.

7.2 Saran

Dari kesimpulan pada penelitian ini didapatkan beberapa saran untuk pengembangan penelitian selanjutnya, antara lain :

1. Agar modul wireless bisa berkomunikasi antar *node* dengan jarak yang cukup jauh dapat menambahkan antena pada modul nRF24L01 agar jangkauan pengiriman data antar *node* bisa lebih jauh.
2. Jika *node* sensor berada pada area yang jauh dan banyak penghalang, agar data bisa sampai pada *node receiver* bisa memperbanyak *node relay* agar data yang dikirim ke *node receiver* bisa sampai.



DAFTAR PUSTAKA

- Arduino, 2017. *Datasheet Arduino Nano*. [online] Tersedia di: <<https://www.arduino.cc/en/Main/arduinoBoardNano>> [Di akses pada 23-06-2017].
- Doxygen, 2016. *Newly Optimized RF24Network Layer*. [online] Tersedia di: <<http://tmrh20.github.io/RF24Network>> [Diakses pada 25-09-2017].
- Hendradjaya, B., Hulu E. 2015. *Tinjauan Penggunaan Jaringan Sensor Nirkabel untuk Pemantauan Gunung Api di Indonesia*. Teknik Elektro dan Informatika, ITB. [online] Tersedia di : <https://www.researchgate.net/publication/277892798_Tinjauan_Penggunaan_Jaringan_Sensor_Nirkabel_untuk_Pemantauan_Gunung_Api_di_Indonesia> [Diakses pada 23-05-2017].
- Iqbal, M. 2015. *Rancang Bangun Wireless Sensor Network Berbasis Topologi Tree-Like Mesh Untuk Sistem Pemantauan Polusi Udara*. [online] Tersedia di: <<https://repository.ipb.ac.id/handle/123456789/76902>> [Diakses pada 23-06-2017].
- Nordic Semiconductor, 2007. *nRF24L01 Single Chip 2.4GHz Transceiver*. [online] Tersedia di : <https://www.nordicsemi.com/eng/content/download/2730/34105/file/nRF24L01_Product_Specification_v2_0.pdf> [Diakses pada 22-09-2017].
- Saputra, M.S.N. 2015. *Impelementasi Arduino dan NR24L01 Sebagai Modul Wireless Sensor Network Menggunakan Metode Master-Slave*. S1. Universitas Brawijaya.
- Satyawan, Arief Suryadi. 2010. *Pembuatan Sistem Jaringan Sensor Nirkabel Untuk Pengukuran Kondisi Lingkungan Pada Medan Terpencil*. Lembaga Ilmu Pengetahuan Indonesia.
- Syafril, D., Wijaya Y. P., Madona P. 2013. *Penghematan Daya Pada Sensor Node Menggunakan Metode Pengaturan Waktu Kirim Data*. [online] Tersedia di: <https://www.researchgate.net/publication/268926119_Penghematan_Daya_Pada_Sensor_Node_Menggunakan_Metode_Pengaturan_Waktu_Kirim_Data> [Diakses pada 20-05-2017].
- Sugiarto, B. 2010. *Perancangan Sistem Pengendali Suhu pada Gedung Bertingkat dengan Teknologi Wireless Sensor Network*. [online] Tersedia di : <<https://ojs.unud.ac.id/index.php/jem/article/view/2321/1528>> [Diakses pada 17-01-2018].
- Suhada, A. 2016. *Sistem Keamanan Gedung Berbasis Wireless Sensor Network Dengan Modal NRF24*. [online] Tersedia di: <http://repository.telkomuniversity.ac.id/pustaka/files/116852/jurnal_eproc/sistem-keamanan-gedung-berbasis-wireless-sensor-network-dengan-modul-nrf24.pdf> [Diakses pada 12-03-2018].